Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

A User's Guide to a Length Frequency Simulator for Exploited Fish Populations

M. C. Rasmussen and R. D. Stanley

Department of Fisheries and Oceans
Biological Sciences Branch
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

September 1988

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1642

Fisheries and Oceans Pêches et Océans

Canada

89-02362/

Canadian Technical Report of Fisheries and Aquatic Sciences

Rapport technique canadien des sciences halieutiques et aquatiques

Canadian Technical Report of
Fisheries and Aquatic Sciences No. 1642

September 1988

A USER'S GUIDE TO A LENGTH FREQUENCY SIMULATOR FOR EXPLOITED FISH POPULATIONS

by

M. C. Rasmussen and R. D. Stanley

Department of Fisheries and Oceans
Biological Sciences Branch
Pacific Biological Station
Nanaimo, British Columbia V9R 5K6

(c)Minister of Supply and Services Canada 1988

Cat. No. Fs 97-6/1642E ISSN 0706-6457

Correct citation for this publication:

Rasmussen, M. C. and R. D. Stanley. 1988. A user's guide to a length frequency simulator for exploited fish populations. Can. Tech. Rep. Fish. Aquat. Sci. 1642: 121 p.

TABLE OF CONTENTS

	Page
ABSTRACT	iv
1. INTRODUCTION	1
2. DESCRIPTION OF THE SIMULATOR	1
3. ORGANIZATION OF THE PROGRAM	6
4. DESCRIPTION OF THE PROGRAM INPUT FILE	7
5. DESCRIPTION OF OUTPUT FILES	9
6. RUNNING THE PROGRAM	11
7. BASIC TROUBLESHOOTING	13
8. ACKNOWLEDGMENTS	14
9. REFERENCES	14
10. APPENDICES:	
A. List of symbols in the COMMON blocks	15
B. Named constants in PARAMETER statements	16
C. Subprograms in File LFMODEL.FOR	17
D. Subprograms in File MORATE.FOR	22
E. Subprograms in File RAND.FOR	28
F. Examples of the Input File	29
G. Examples of the Output Files	30
H. Examples of Terminal Display for an Interactive Session	58
I. Program Listing for File LFMODEL.FOR	72
J. Program Listing for File MORATE.FOR	98
K. Program Listing for File RAND.FOR	118

ABSTRACT

Rasmussen, M. C. and R. D. Stanley. 1988. A user's guide to a length frequency simulator for exploited fish populations. Can. Tech. Rep. Fish. Aquat. Sci. 1642: 121 p.

A user's guide is presented for a simulator which generates a predicted length frequency distribution for an exploited fish population. The simulator first calculates available numbers at age from estimates of the instantaneous fishing and natural mortality rates, and partial recruitment factors. It then randomly selects individuals of known age from the age distribution and assigns a length to each individual according to the input growth formula. Users can select options for stochastic variation in recruitment, mean length at age, and individual length at age, as well as choosing between a von Bertalanffy or Schnute growth parameterization.

RESUME

Rasmussen, M. C. and R. D. Stanley. 1988. A user's guide to a length frequency simulator for exploited fish populations. Can. Tech. Rep. Fish. Aquat. Sci. 1642: 121 p.

On présente un guide d'utilisateur pour un simulateur qui génère la distribution prévue de la fréquence des longueurs pour une population de poissons exploitée. Le simulateur calcule d'abord la population disponible par âge à partir d'évaluations du taux de mortalité instantanée par pêche et du taux de mortalité naturelle, et des facteurs de recrutement partiel. Il choisit ensuite au hasard des individus d'âge connu à partir de la distribution d'âge et assigne une longueur à chaque individu selon la formule de croissance à l'entrée. Les utilisateurs ont la possibilité de choisir des variations stochastiques en ce qui concerne le recrutement, la longueur moyenne par âge et la longueur individuelle par âge, ainsi que de choisir entre la formule de paramétrisation de la croissance de Bertalanffy ou celle de Schnute.

1. INTRODUCTION

The simulation model described herein was developed from the length frequency simulator of Breen and Fournier (1984) as adapted by Stanley (1987). The model uses input information on instantaneous natural and fishing mortality, partial recruitment, and length at age to generate theoretical length frequency distributions. Options include stochastic variation in recruitment and mean or individual size at age. Users can also choose between the growth parameterizations of von Bertalanffy (Ricker 1975) or Schnute (Schnute 1981).

This document contains a description of the model and options, explanations and examples of the input files, operating instructions, output files, and a copy of the FORTRAN code. Requests for a version on tape or via direct transmission should be directed to the junior author. The simulator has proven useful for assessment when catch-at-age time series or catch rate indices are not available, but when length frequency samples have been collected (Stanley 1987).

2. DESCRIPTION OF THE SIMULATOR

2.1. RELATIVE ABUNDANCE AT AGE

The model has two modules. The first generates relative abundance at age in the fishable population. The second module samples a user-specified number of fish from that age distribution and assigns a length to each.

If:

M_k = instantaneous natural mortality rate at age k, (k=1,....r)
R_k = partial recruitment factor at age k, (k=1, r;)

where r is the maximum age of a cohort, and

F_j = instantaneous fishing mortality rate in year j, (j=1,...s)

where s is the most recent year, then the total instantaneous mortality (Z) for cohort i in year j at age k can be expressed as

$$Z_{ijk} = (F_j R_k) + M_k. \qquad (1)$$

Total instantaneous mortality (D_{ij}), experienced by cohort i since recruitment to the present year j, equals the summation of (1) over all years of the i^{th} cohort's life:

$$D_{ij} = \sum_{k=1}^{r} [(F_{j-r+k} R_k) + M_k]. \quad (2)$$

[Note: for computational convenience the simulator actually treats the present year as year 1 for the vector of fishing mortality rates and works backwards over time.]

The relative abundance of the i^{th} cohort in year j can be represented by

$$P_{ij} = e^{-D_{ij}}. \quad (3)$$

Applying a random deviate, τ , and the partial recruitment parameter, R_k, to reflect the lower likelihood of catching the incompletely recruited cohorts, the relative abundance of each cohort in the catch in year j becomes

$$P_{ij} = R_k \, e^{(V-D_{ij})} \quad (4)$$

where $V = \sigma_x * \beta_y$. σ_x is the standard deviation in recruitment and β_y is a normally distributed random number, with a mean of 0 and a standard deviation of 1. The random number series is initialized when calculating the relative abundance of the first age class.

2.2. MEAN LENGTH AT AGE

Mean length at age is calculated from a von Bertalanffy or a Schnute growth formulation with an option for stochastic variation.

2.2.1. von Bertalanffy

Mean length, Y(k) for the k-th age class, is given by

$$Y(k) = Y_\infty(1-e^{-g(k-t_0)}), \quad (5)$$

where Y_∞ is the asymptotic length of the species, g is the von Bertalanffy growth coefficient ("k" in the original notation), and t_0 is the hypothetical age at zero size. Note that g is sometimes called the Brody coefficient and can be expressed by

$$g = - \log_e(Fr), \quad (6)$$

where Fr is the Ford coefficient (Ricker 1975).

2.2.2. Schnute

The mean length at age with the Schnute model is given by

$$Y(k) = y_1^b + (y_2^b - y_1^b)\left(\frac{1-EXPA\ (k-\tau_1)}{1-EXPA\ (\tau_2-\tau_1)}\right)^{1/b} \qquad (7)$$

where y_1 and y_2 are the mean lengths of the species at ages τ_1 and τ_2 respectively, ($y_2 > y_1 > 0$, $\tau_2 > \tau_1$), k is the age of the itn cohort, and EXPA and b are growth coefficients.

For the above formulation, EXPA must not be equal to one, and b must be nonzero. Schnute refers to this as Case 1 of his formulation. This expression differs from that in Schnute (1981) in that we have replaced e^{-a} in the original expression by EXPA. EXPA and "a", therefore, have the same the relationship between them, as the Ford and Brody coefficients have in the von Bertalanffy expression. The expressions for the 3 other cases where EXPA=1 and/or b=0 are shown below:

Case 2. EXPA not equal to 1, b=0:

$$Y(k) = y_1 \exp\left(\log(y_2/y_1)\ \frac{1-EXPA(k-\tau_1)}{1-EXPA(\tau_2-\tau_1)}\right), \qquad (8)$$

Case 3. EXPA=1, b not equal to 0:

$$Y(k) = \left(y_1^b + (y_2^b - y_1^b)\ \frac{k-\tau_1}{\tau_2-\tau_1}\right)^{1/b} \qquad (9)$$

Case 4. EXPA=1, b=0:

$$Y(k) = y_1 \exp\left(\log(y_2/y_1)\ \frac{k-\tau_1}{\tau_2-\tau_1}\right) \qquad (10)$$

If a growth rate can be expressed by the von Bertalanffy model, it can be expressed by the Schnute model with the Schnute coefficients determined as follows:

i) EXPA is determined from: EXPA = e^{-g} or EXPA = FR;

ii) b=1 for von Bertalanffy generalized growth (Pütter No.1);

iii) The values of y_1 and y_2 can be calculated from equation (4), by using ages at opposite ends of the von Bertalanffy growth curve.

2.2.3. Application of a Random Deviate to Mean Length at Age

The mean lengths calculated above are varied stochastically by applying a random deviate as follows:

$$L(k) = Y(k) + \delta_Y * RAND \qquad (11)$$

where L(k) is the new mean length at age k, δ_Y is the standard deviation of the mean length about the predicted mean length, and RAND is a normally distributed random number, with a mean of zero and a standard deviation of 1. The series is re-initialized when calculating the mean length of the first age class.

2.3. GENERATING A LENGTH FREQUENCY DISTRIBUTION

Individual fish of known age are randomly selected from the age distribution with the likelihood of selection in proportion to abundance at age. The length, ℓ, for a fish placed from the kth age is determined by

$$\ell v = L(k) + \delta_y * RAND \qquad (12)$$

where L(k) is the mean length corresponding to the k^{th} age, δy is the standard deviation in individual lengths about mean lengths. This random number series was also re-initialized when determinating the age of the first fish.

2.3.1. Length distribution of the sample

The length distribution is determined by placing individual fish in length categories of a specified width. The range of lengths to include in the distribution is determined by

$$L1 = L(1) - \delta_y w \qquad (13)$$

$$L2 = L(n) + \delta_y w \qquad (14)$$

where L1 and L2 respectively, define the lengths of the minimum and maximum categories. L(1) and L(n) are the mean lengths of the youngest and oldest age classes. δ_y is the standard deviation in individual length about mean length, and w is a parameter to specify the possible range of the length distribution. L1 is also rounded down to an integer length, while L2 is rounded up. If w is small or zero, the simulator will generate fish that are either too small or too large for the distribution categories. It is used to restrict lengths to a realistic range.

The number of length categories, N1, is given by

$$N1 = \frac{L2-L1}{W} + 1 \qquad (15)$$

where W is the width (i.e. 1 cm.) specified for the length categories in the length distribution.

2.3.2. Statistics for the Sample Population

The mean length, median length (to the nearest half-interval), and the standard deviation in the lengths are determined for the sample population. Fish excluded from the length distribution (because of a small w) are counted, but not included in the length statistics.

3. ORGANIZATION OF THE PROGRAM

The program has three units. The first includes a main program to call the length frequency simulator subroutine, and utilities to call files and print the results of the simulation. It also includes the length frequency simulator subroutine NASIM and the utilities subroutines used by the above. These routines are in file LFMODEL.FOR (Appendices C and I).

The second unit is module MORATE, and consists of a collection of interactive routines to enter and/or edit the mortality schedule. This module also provides a cumulative total mortality rate for each cohort. These routines are in file MORATE.FOR (Appendices D and J).

The third unit is function RAND, an optional version of a uniform random number function to be used as an alternative to either the International Mathematics and Statistics Library's Version 9.0 (IMSL) function GGUBFS or the VAX mathematical library function RAN. This function appears suitable and has the advantage of being portable among a wide variety of computers. This routine is included in file RAND.FOR (Appendices E and K).

Summaries of the COMMON blocks and PARAMETER statements are also included (Appendices A and B). These will be useful if the user needs to change array dimensions, file names or logical unit numbers.

4. DESCRIPTION OF THE PROGRAM INPUT FILE

The driving parameters are read in free format by subroutine NASIM from unit INUNIT on file INFILE. For the present version of the program, this file is LFMODEL.DAT on logical unit 10. The file is described below; two examples are shown in Appendix F.

Record 1:

TITLE - Title, maximum of 80 characters.

Record 2:

NFISH - Sample size (Integer).

Record 3:

NAGE - Number of age classes (Integer).
MINAG - Age of recruitment (Integer).
LINT - Size of length interval (Real). Print routines currently assume that value will be greater than or equal to 1.0 cm.
FLAG - Constant length used in the lineprinter plot as a visual reference (Real). It will not be flagged if FLAG is outside the range simulated.

Record 4: von Bertalanffy constants

VBL - Length at k = ∞ (Real).
VBK - Growth constant (Brody coefficient) (Real).
VBT - Extrapolated age at zero size (Real).

Record 5: Schnute constants

Y1 - Size at age TAU1 (Real).
Y2 - Size at age TAU2 (Real).
EXPA and B - growth constants (Real).
TAU1 - Age at size, Y1 (Integer).
TAU2 - Age at size, Y2 (Integer).

Record 6:

VVB - Standard deviation of mean lengths about predicted mean length (Real).

Record 7:

SDLAGE - Standard deviation of individual lengths about mean length (Real).

Record 8:

RANGE - The number of standard deviations (in record 7) to extend the length range beyond the largest and smallest mean lengths (Real). If RANGE is too small, the simulation will produce lengths that will be beyond the upper and lower limits for the distribution. These will be excluded from the summary statistics.

ISEEDP - The random seed for calculation of the population in each cohort (Integer). This should be a positive, large, odd-valued integer, up to 9 digits long. See the note below, in reference to ISEEDL.

ISEEDL - The random seeds used for calculating the mean size at age. This should be a positive, large, odd-valued integer, up to 9 digits long. See the note below. The two initial seeds used to generate random numbers should be positive valued integers. For a 32-bit computer (or for 16-bit machines where INTEGER*4 variables are allowed), this requires that these numbers be within the range of 0 to +2,147,483,647. The program ensures that the seeds are odd-valued (preferred for VAX math library's random number routine).

Record 9:

ZFMORT - The default for the fishing mortality rate (Real). Any value of ZFMORT is permitted, even though the interactive portion of this program will not allow the user to enter rates greater than 3.0 or negative rates.

ZNMORT - The default for the natural mortality rate (Real). Any value of ZNMORT is permitted, even though the interactive input will not allow the user to enter rates greater than 3.0 or negative rates.

RSIG - The standard deviation to use when applying the recruitment parameters (Real). This permits stochastic variation in recruitment.

Record 10:

PREC(I) - First 10 partial recruitment parameters (Real).

Record 11:

PREC(I) - Remaining partial recruitment parameters from the 11th age after recruitment to the maximum age (Real).

5. DESCRIPTION OF THE OUTPUT FILES

All files contain a FORTRAN carriage control character (a blank or a "1") in the first column, to produce a more readable copy if the file is printed.

5.1. RESULTS OF THE SIMULATION

The results are written to one of a list of files defined by subroutine ASSIGN. Logical unit IUNITO is used. The list of file names is searched for a file (LFMFILE.OU*) that does not currently exist on the computer. The first available name is used for the output from the current simulation. When only one "new" file name can be found, the user is warned that the list of output files will be depleted after this run. If there are no suitable file names, the user is informed, and the program stops (Appendix G). At present, logical unit 10 is used for the output of the results from the simulations. The five file names currently defined are:

Order of Search	File Name
1	LFMFILE.OU1
2	LFMFILE.OU2
3	LFMFILE.OU3
4	LFMFILE.OU4
5	LFMFILE.OU5

5.2. SUMMARY OF LENGTH DISTRIBUTION FOR GRAPHING

A data summary is written to a second file as well as to the "results" file. If the file already exists, the data are appended to the previous simulation. This file will then contain the distribution summaries for up to 5 simulation runs (MAXSAV in subroutine DSSAVE), which can then be used as input to a graphics routine. When the fifth data set is written, the user is sent a warning that this data set may be overwriting a previously written fifth data set. These concatenated distribution data sets are written to LFMFILE.PRO on unit 21.

If an error occurs when reading any previously written data, the current data set is written to another file, DMPFLE. A warning is sent to the user that this alternate file is being used. The file is defined as LFMFILE.DMP on unit 22.

Each data set is prefaced by a 12-line header to briefly describe the conditions for the simulation. The header includes the format of the distribution, run number, the "results" file name to which this distribution

corresponds, etc. The first data set to the file has a run number of "1". Subsequent run numbers will always be one greater than the one preceding it in the file, even if data sets are removed from the file.

The data set header is also written to the "results" file. This allows the user to match the "results" file to its summary file (see Appendix G).

The summary file and all others in this program were opened using standard FORTRAN 77, OPEN statement options. The VAX extension to the language to allow CARRIAGECONTROL='LIST' was not used. If the output from this program is to be used as input to a SAS routine on a VAX installation, the files will need to be converted to a CARRIAGECONTROL='LIST' type of file.

5.3. MORTALITY RATE SCHEDULES

The module MORATE also creates files. The current mortality schedule is always saved. This file is updated for each simulation so it may be used only by the next simulation. It is written to file MFLLAS on unit MUNLAS. At present, it is named LFMORTRT.LAS and is on unit 11.

The user may also create a more permanent copy of the schedule. It is updated only when the user specifically requests it. The file MFLSAV on unit MUNSAV is used for this explicitly saved mortality schedule. At present, the file is named LFMORTRT.SAV and is on unit 12 (see Appendix G).

6. RUNNING THE PROGRAM

The model runs interactively with the following options:

a) Choice of von Bertalanffy or Schnute growth functions. The parameters are defined in LFMODEL.DAT.
b) Choice of the fishing (F) and natural mortality (M) rates from the following:
   i) Use the rates defined for the preceding run; or
   ii) Use a constant M for all age classes, and a constant F for all years, where the user may specify the rates or use the defaults defined in file LFMODEL.DAT, or
   iii) Use a schedule where rates are specified for a range of age classes, over a range of calendar years from the present, back in time to when the oldest fish were of age of first recruitment. The user may change or extend an existing schedule, or create a new schedule. The fishing and natural mortality rate schedules are editted separately. Since a schedule must not contain overlapping or duplicate time periods, items can be deleted by specifying a new time period greater than or equal to the period for all of the items to be removed.

The program always retains the schedule. It may be used as is, as in option i) above, or edited. The user may explicitly save the schedule for use in future runs of the program. For any run where variable mortality has been requested, the program may run on either the preceding run's schedule or on an explicitly saved schedule from a previous run.

A default schedule, equivalent to the default constant mortality rates, will be used if the user requests a schedule that has not been previously defined. This will be the case when the program is run for the first time, or when the schedule files have been deleted or renamed.

6.1. INTERACTIVELY ENTERING MORTALITY RATES

The mortality rates are read in F10.0 format. If the decimal is not entered, then a whole number is assumed (e.g. 2 becomes 2.0). A decimal point without a number will be interpreted as a zero (0.0). Values greater than 3.0 and less than 0.0 are not permitted.

The rates ZFMORT and ZNMORT, read by subroutine NASIM (see 4.0), are used as defaults for the mortality rates. They can be used as defaults for constant mortality or for variable mortality when creating a new schedule or extending an existing one. Their values are not restricted and can be greater than 3.0 if required.

6.2. INTERACTIVE INPUT OF YEAR/COHORT MORTALITY RATES

The ranges of years for entering mortality rates are read in I10 format. Values must be greater than zero. The default value for the range in a new schedule is the larger of either 150 (DEFLTY set in subroutine MORVRO) or the number of age classes for this simulation (NAGE in the input file). If variable rates are being used, but the schedule does not cover all age classes, the last rate (i.e. most ancient) is extended to cover these older age classes.

When entering the mortality rate for one year (n) the range is specified as n to n, not n to n+1. For example if the mortality rate for age 3 is entered, the range is 3 to 3.

6.3. SUMMARY OF MORTALITY RATE DEFAULT VALUES

i) Constant rates:
   Fishing mortality - ZFMORT, 1 to NAGE
   Natural mortality - ZNMORT, 1 to NAGE.

ii) Rate schedule from previous run does not exist:
   Fishing mortality - ZFMORT, 1 to larger of NAGE or 150
   Natural mortality - ZNMORT, 1 to larger of NAGE or 150

iii) An explicitly-saved schedule does not exist:
   Fishing mortality - ZFMORT, 1 to larger of NAGE or 150
   Natural mortality - ZNMORT, 1 to larger of NAGE or 150

iv) Extending an existing schedule:
   Last rate in the schedule is applied for one year beyond the previous item's year or to the year NAGE, whichever is the larger.

v) Editting an existing schedule:
   By default, the rate and range are not changed.

7. BASIC TROUBLESHOOTING

7.1. WARNING FOR THOSE UNFAMILIAR WITH PARAMETER NAMED CONSTANTS

While PARAMETER named constants have many advantages (easy re-dimensioning of a program, documentary style of specifying constants, etc), they must be used with care. They must not be used directly in COMMON blocks or in subroutine calls. They are constants that are set at compile time and are to be used only in their originating subprogram. If the value of such a constant is to be passed to other routines, the value must first be assigned to a regular variable. This restriction is to prevent the inadvertent redefinition of a constant used in a COMMON block or in a subroutine call.

Many compilers check for such a violation to the FORTRAN 77 standard. The VAX compiler (version 4.0) checks only for its use in COMMON blocks. A VAX runtime error (%SYSTEM-F-ACCVIO, access violation) usually occurs when PARAMETER named constants are put into subroutine calls.

7.2. PORTABILITY PROBLEMS WITH TAB CHARACTER IN THE SOURCE CODE

The current version of this program does not contain TAB characters in the source code. This was done to keep the code more system independent. For VAX systems, TAB's are interpreted as a move to the next column out of columns 8, 16, 24, 32, 40, etc. The VAX compiler correctly interprets these positions. However, compilers for most other computers will not handle tabs. Most microcomputers will not handle TABs in the code, so be sure to remove any tabs that you may have entered before creating a "micro" version of this program.

8. ACKNOWLEDGMENTS

We would like to thank Koit Teng, Bob Johns and Len Smith of the Computing Services staff of the Institute of Ocean Sciences, Sidney for their assistance and for making their facilities available for this project, we also thank Bruce Leaman and Jeff Fargo for reviewing the manuscript.

9. REFERENCES

Breen, P. A. and D. A. Fournier. 1984. A user's guide to estimating total mortality rates from length frequency data with the method of Fournier and Breen. Can. Tech. Rep. Fish. Aquat. Sci. 1239: 63 p.

Ricker, W. E. 1975. Computation and interpretation of biological statistics of fish populations. Fish. Res. Board Can. Bull. 191: 382 p.

Schnute, J. 1981. A versatile growth model with statistically stable parameters. Can. J. Fish. Aquat. Sci. 38(9): 1128-1140.

Stanley, R. D. 1987. Use of a frequency simulator to explore the information content in length data for long-lived species, silvergray rockfish (Sebastes brevispinis). Proc. Int. Rockfish Symp., Oct. 1987. Anchorage, Alaska.: 155-170.

APPENDIX A. List of Symbols in the COMMON blocks

Two common blocks are used to transfer information between subroutines. The number of variables in the common blocks has been kept small by including only those parameters shared by several subroutines. Most subroutines use calling sequences to pass information. This was done to define the areas where variables are actually needed and simplify future program development. The common block LGUNIT is used in most subroutines to define logical units for interactive I/O:

INTKEY --- logical unit for interactive input
INTCRT --- logical unit for interactive display
IUNITO --- logical unit for output from simulation.

The common block OVBLK is used for variables that drive the simulator. Its variables are:

Integers scalars:

MAXAG --- maximum age in simulation
MINAG --- minimum age in simulation
MXFREQ --- limit to the number of intervals
MXYRS --- limit to the number of years/ages
NAGE --- number of cohorts to simulate
NFREQ --- number of length frequencies
NLONG --- number of fish longer than lengths of interest
NSHORT --- number of fish shorter than lengths of interest.

Real scalars:

L1 --- smallest expected length
L2 --- largest expected length
LINT --- interval for distribution
POPORG --- normalization factor before recruitment
POPSUM --- normalization factor after recruitment
RSIG --- standard deviation in recruitment
VBK --- von Bertalanffy k constant (Brody coefficient).

Real arrays:

F --- length frequency distribution
M --- mean length at age
P --- relative number at age after recruitment
PREC --- partial recruitment parameters
S --- standard deviation of length at age
SAMORT --- sum of the annual mortality rates from MINAG up to the maximum age of the cohort
FYEAR --- fishing mortality rate for each year from the present back to when the oldest cohort was recruited
MAGE --- natural mortality rate for each age.

Character scalars:

TITLE --- (80 characters, maximum).

APPENDIX B. Named constants in PARAMETER Statements

Parameter-named constants dimension most arrays. The program can be redimensioned by changing only the parameter statements. They also define I/O units and corresponding file names. Some that may need to be changed are described below. Parameters used to dimension the arrays are set in the mainprogram and in subroutines LPOUT, LPPLOT, and NASIM (current default in brackets):

MAXFRQ -- the maximum number of length intervals (100)
MAXYRS -- the maximum number of years (i.e. age classes) (150).

Main I/O units are set in the mainprogram:

IKEY -- logical unit for interactive keyboard input (5)
ICRT -- logical unit for interactive display (6)
IOUT -- logical unit for output from simulation (20).

Number of different output file names set in subroutine ASSIGN:

NFILEO -- number of files available for output (5)

The files for data summaries are set in subroutine DSSAVE:

MAXSAV -- maximum number of data sets to save per file (5)
IUNTP -- unit number of file for concatenated data (21)
IUNITD -- unit number of file for "dumping" a single data set if unable to use IUNTP (22)
PRFILE -- filename for concatenated sets ('LFMFILE.PRO')
DMPFLE -- name of file to use if "dumping" a data set ('LFMFILE.DMP').

The format for the lineprinter plot is set in subroutine LPPLOT:

IWIDTH -- the number of columns for the 'plot' width (50)
FPOS -- .TRUE. if function to plot is all positive. If .False., then will use only half of IWIDTH for the positive portion of the plot.

The input file that controls the simulation is set in NASIM:

INFILE -- input file ('LFMODEL.DAT')
INUNIT -- unit number of the input file (10).

The mortality rate schedule used in module MORATE is set in subroutine MORATE:

MXSCHD -- maximum number of periods in schedule (50)
MFLLAS -- file containing the default schedule as used for the LAST run ('LFMORTRT.LAS')
MFLSAV -- file to contain the explicitly SAVED schedule ('LFMORTRT.SAV')
MUNLAS -- logical unit to use for MFLLAS (11)
MUNSAV -- logical unit to use for MFLSAV (12).

APPENDIX C. Subprograms in File LFMODEL.FOR

LFMODEL.FOR generates length frequencies. It calls subroutines to:

1. Assign the appropriate file for output;
2. Simulate age and length compositions;
3. Print message if any lengths outside range of interest;
4. Print results;
5. Plot the length-frequency distribution on a line-printer;
6. Save the distribution for this run concatenated to the data sets saved from previous runs. Write message to OUTFLE, to describe the file used;
7. Close output file.

SUBROUTINE ASSIGN(OUTFLE,LASTFL)

Assigns a file name to OUTFLE for printed output. A list of available file names is defined and scanned. The first name that will be a new file is assigned to OUTFLE and is opened. If all possible files already exist, program stops, sets the logical flag, LASTFL, to true if all output files will contain data after present run. Has an entry point, ASSGCL(OUTFLE,LASTFL), to close the file OUTFLE opened by the call to ASSIGN. If the flag LASTFL is set when ASSGCL is called, then a message is written to indicate that the output file "stack" will be full after run.

SUBROUTINE DSSAVE(LINT,L1,L2,F,MTYPE,OUTFLE,NFREQ)

Saves the NFREQ values of the distribution, F, for the length range of L1 TO L2, at intervals of LINT. The distribution for this run is concatenated to the distribution from the preceding run with a descriptive header. The header includes the type of mortality schedule (MTYPE), and the name of the main output file (OUTFLE). The subroutine also defines an alternate file to use in case an error occurs while trying to concatenate distributions.

SUBROUTINE FGEN(P,NAGE,IAGE,M,S,IL,NFREQ,IX0,L1,L2,L,LINT,
NLONG,NSHORT)

Generates a fish of known age, IAGE, and length, IL. If the length is too long or short for the range of the length categories, IL = -1.

SUBROUTINE GETMS(IXO,M,MINAG,NAGE,S,SDLAGE,VVB,VONBRT,VBK,VBL,VBT,
A,B,ITAU1,ITAU2,Y1,Y2)

Defines the standard deviation and mean lengths for each age. If VONBRT is true, it will use von Bertalanffy's growth function (also called Pütter No. 1). If false, it will use the Schnute model. The mean lengths at age are altered by a random normal fraction (mean of zero) of the standard deviation in mean length.

SUBROUTINE GETPS(NAGE,IXO,SAMORT,RSIG,P,PREC,POPORG,POPSUM)

Calculates the relative number of fish in each cohort using SAMORT(I), the accumulated mortality rate that the i^{th} cohort has experienced since recruitment. SAMORT(I) is defined in subroutine MORSUM. More specifically it:

1. Determines two pseudo-random numbers in the range of 0 to 1.0 and from these determines a random number from a normal distribution with a mean of zero. A random normal deviate is determined from the random number times the specified standard deviation in the recruitment.
2. Calculates the relative population of the i^{th} cohort from the negative exponential of the accumulated mortality (SAMORT) varied by the random normal deviate.
3. Applies partial recruitment factors to relative population.
4. Calculates factors to normalize the original and recruited populations.
5. Normalizes the recruited population of the k^{th} age class.

SUBROUTINE HOLD(INTCRT,INTKEY)

Holds the terminal screen until user responds with a carriage return. Allows the interactive user to view messages displayed during the execution of the program.

SUBROUTINE LPOUT(OUTFLE)

Prints the age class results from the simulation.

SUBROUTINE LPPLOT(FLAG,NFISH)

Plots length distributions as a printed page to be viewed on a terminal or lineprinter at least 80 characters wide. Output is to unit INTCRT (i.e. terminal if an interactive run) and to unit IUNITO. Draws a line on the abscissa at the value of "FLAG" to act as a reference. Line is drawn only if "FLAG" is within the range of the data.

SUBROUTINE NASIM(FLAG,MTYPE,NFISH)

Principal subroutine to control the length-frequency simulation. Reads information from INFILE on unit INUNIT to define the length simulation. The coefficients for both are supplied from LFMODEL.DAT. The default is the von Bertalanffy formulation. Also determines the range for the length frequency distribution and the number of frequencies. Calculates the mean, median, and standard deviation of the length distribution. Controls simulation by reading input parameters from LFMODEL.DAT (unit 10) and then calls MORATE, GETPS, GETMS, and FGEN in order.

The random number series used in subroutines GETPS, GETMS and FEGEN, are initialized with the following integer-valued seeds. The variable, IX0 (real*8) is used to retain the value of the seed as updated by the random number function. The values of ISEEDP and ISEEDL should be large, odd-valued integers):

ISEEDP --- initial seed for population of cohorts
ISEEDL --- initial seed for mean lengths
ISEEDI --- initial seed for ages and lengths of individual fish (ISEEDI is the mean of ISEEDP and ISEEDL)
IX0 --- retains successive values of the seed used by the pseudo-random number function.

SUBROUTINE NASMCK(L1,L2,NLONG,NSHORT)

Checks if NASIM generated fish either too long or too short for the lengths of interest. Prints message if any found.

SUBROUTINE PARERR(MAXFRQ, MXFREQ, MAXYRS, MXYRS)

Prints error message if parameter MAXFRQ does not equal variable MXFREQ or when parameter MAXYRS does not equal variable MXYRS.

FUNCTION RANNUM(IX0)

A 'dummy' function to provide pseudo-random real numbers in the range of 0.0-1.0. IX0 is the external value of the seed used by the random number function subprogram. IX0 is updated if RAN or GGUBFS is called, but not if using RAND. For compatibility with any of the possible functions that could be called below, IX0 should be declared REAL*8 in the routine that calls RANNUM.

RANNUM allows the programmer to choose the function for providing the pseudo-random numbers. Each routine expects a different form for the seed (see below). The appropriate variable is therefore assigned the value of IX0 prior to calling the function subprogram. The choice of the function is from the statements below, where the desired routine is active and the rest are commented-out. This method of choosing a function, while not elegant, does not invoke unnecesary libraries and will likely be done only when using an alternate computer system. The present choice is from the following routines:

RAN(IXI) --- a VAX Fortran math library function where IXI is an INTEGER*4 seed updated by the function. IXI should be set initially to a large, odd-valued integer (less than +2,147,483,647) (via IXI = INT(IX0)).

GGUBFS(IXD) -- An International Mathematics and Statisics Library (IMSL) function. IXD is a REAL*8 seed assigned an integer value by the function. IXD should be set initially to an integer value in the inclusive range of 1.D0 to 2,147,483 647.D0 (via IXD = IX0). (This library routine can be included when LINKing via: $LINK <program>,sys$SHARE:IMSLIBS/lib for the VAX at PBS, Nanaimo, or $LINK <program>,ccs$IMSLLIB/lib for the VAX at IOS, Sidney)

RAND(RSEED)-- a subroutine obtained from the SLATEC public domain library for MS-DOS personal computers (see file RAND.FOR). RSEED is the initial seed or a flag to control the random number function (set via RSEED = IX0/(2**22). If RSEED is positive, then the top two 11-bits of the fractional part of RSEED will be the seed. If RSEED is zero, then the next pseudo-random in the series will be generated. If RSEED is negative, the last generated number will be returned and usable as the seed to continue the series.

A message is printed to the output file to indicate which random number routine is being used. To allow for the different seed wanted by function RAND, for its first call, the seed IX0 is divided by 2**22 (i.e. 4194304). RAND will then update the seed internally and maintain its value between calls.

SUBROUTINE SCHNTE(EXPA,B,NAGE,TAUS,TAU1,TAU2,Y1,Y2,Y)

This subroutine produces an array of length Y for NAGE age classes of fish, with the Schnute parameterization (see 2.3.2).

The coefficients correspond to Schnute's paper, except for "EXPA" which equals the exponential of the negative of his coefficent, "a". This change was made to reduce the computational time.

The growth function above, is used when EXPA is not equal to one and B is not equal to zero. There are also three alternate formulations for Schnute's special cases where a=0 (i.e. EXPA=1) and/or b=0.

APPENDIX D. Subprograms in MORATE.FOR

MORATE.FOR is a collection of subroutines to interactively define mortality rates. All of its subroutines have 6-letter names that begin with MOR. The mainprogram calls MORATE from subroutine NASIM. Subroutine MORATE either directly, or indirectly, calls the other subroutines in this module. It is an interactive program module to do two functions: (1) read and/or edit a mortality schedule, and (2) sum annual rates for each cohort. The subprogram operates in 3 basic modes:

1) uses the schedule of the previous run;
2) uses constant rates for fishing and natural mortality;
3) uses rates that can vary for each year for fishing mortality, and each age for natural mortality.

Mode 1: Use the previous rate schedule. MORATE calls:

MORVRO -- to open and read a file of mortality rates
MORVWR -- to write a mortality schedule
MORVTR -- to transfer the schedule values to arrays of rates for each year/age-class
MORHLD -- to hold "screen" before printing any more results
MORSUM -- to calculate the cumulative mortality rates

Mode 2: Use a constant mortality rate. MORATE calls:

MORCON -- for a constant annual mortality rate
MORSUM -- to calculate the cumulative mortality rates,

and MORCON in turn, calls:

MORCSV -- to save a schedule of constant rates.

Mode 3: Use variable mortality rates. MORATE calls:

MORVAR -- for variable mortality rates
MORSUM -- to calculate the cumulative mortality rates,

and MORVAR, in turn, calls:

MORZRO -- to initialize rate schedule arrays
MORVRO -- to open and read a file of mortality rates
MORVED -- to edit a schedule
MORVEX -- to extend a schedule
MORVWR -- to write a schedule
MORVTR -- to transfer the schedule values to arrays of rates for each year/cohort
MORVSV -- to save the edited schedule,

and indirectly via MORVED and MORVEX calls:

MORVRR -- to edit the rates
MORVRY -- to edit the ranges of years in a schedule.

SUBROUTINE MORATE(FYEAR,MAGE,MINAG,MAXAG,MTYPE,NAGE,PREC,SAMORT,
ZFMORT,ZNMORT)

a) Mortality rate schedule

Reads a schedule (from units MUNLAS or MUNSAV) and calculates their sums for each age from MINAG to MAXAG. If a schedule does not exist, then one is created where ZFMORT is used for all fishing rates from the present date back NAGE years and ZNMORT is used for the natural mortality rate for all age classes. If a schedule exists, the user has the option of using it for the simulation, either as is or modified. If it is modified, the user also has the option of using the modified schedule to update the one on unit 11, for future use. NOTE:- The schedule must be entered in order of increasingly "ancient" years and older age classes.

b) Sum of annual rates for each age group.

Using the mortality schedule, annual rates are determined for each year covering the age groups of MINAG to MAXAG. These rates are summed for each group from MINAG to the age of the group. These sums are returned as elements of the array SAMORT.

Calling Sequence:

FYEAR	---	fishing mortality rate for each year starting from the present
MAGE	---	natural mortality rate for each age class
MINAG	---	minimum age in the simulation
MAXAG	---	maximum age in the simulation
MTYPE	---	20 character name for type of mortality
NAGE	---	number of ages to simulate
SAMORT	---	array of rates summed over the life of each cohort from MINAG to the age of the group
PREC	---	partial recruitment for each age
ZFMORT	---	default fishing mortality rate
ZNMORT	---	default natural mortality rate.

Note: If variable rates are to be used but a rate schedule does not exist then the value of ZFMORT will be used for all years back to the year when the NAGE age class recruited. ZNMORT will be used for all age classes up to NAGE.

Common block /LGUNIT/ variables:

INTKEY	---	default logical unit for interactive input
INTCRT	---	default logical unit for interactive display
IUNITO	---	logical unit for output from simulation.

Parameters:

LOGKEY	---	default logical unit for interactive input
LOGCRT	---	default logical unit for interactive display
MXSCHD	---	maximum number of periods in the schedule
MFLLAS	---	filename containing the default schedule as used for the LAST run (Filename.type form for VAX)
MFLSAV	---	name of the file which will contain the explicitly SAVED schedule (Filename.type form for VAX)
MUNLAS	---	logical unit to use for MFLLAS
MUNSAV	---	logical unit to use for MFLSAV.

Functions performed from subroutine MORATE:

1. If the schedule from the last run is to be used then:

    a. read rates from last run (default file)
    b. write rate schedule to output unit
    c. transfer the fishing rate schedule values to FYEAR array for each year of the simulation
    d. transfer the natural mortality rate schedule values to MAGE
    e. array for each age class in the simulation
    f. wait until user has read screen.

2. If constant rates, then set rates;
3. If variable rates, then define or edit variable rates;
4. Calculate the cumulative mortality rates for each age class.

SUBROUTINE MORHLD(INTCRT,INTKEY)

Hold screen until user responds with a carriage return. Message written to unit INTCRT. Response is read from unit INTKEY.

SUBROUTINE MORCON(FYEAR,MAGE,MFLLAS,MUNLAS,NAGE,ZFMORT,ZNMORT)

This subroutine sets constant values for the mortality rates. Rates must be positive, and less than or equal to 3.0. If no rate is entered, then the default values are used. The constant fishing mortality rate applies to all years back to when the oldest fish was recruited, while the constant natural mortality rate apply to all age classes. The routine also saves a schedule.

SUBROUTINE MORCSV(CFRATE,CNRATE,MFLLAS,MUNLAS,NAGE)

Saves the constant rate schedule for use in the next session.

SUBROUTINE MORSUM(FYEAR,MAGE,NAGE,PREC,SAMORT)

Calculates SAMORT(I), the cumulative mortality rate of the i^{th} cohort (see 2.1)

$$SAMORT(I) = \sum_{k=1}^{I} PREC(k)FYEAR(I-k+1)$$

$$+ \sum_{k=1}^{I} MAGE(k)$$

The sum of MAGE(K) for k=1 to I, is the natural mortality rate experienced by fish of the i^{th} cohort, from their recruitment age up to their present age. FYEAR(L) is the fishing mortality rate for (L-1) years ago, where L = 1,2,3,...,I and FYEAR(1) is the fishing mortality rate this year, and FYEAR(I) is the rate when the i^{th} age class was recruited. PREC(k) is the partial recruitment parameter for fish of the k^{th} age class.

For example, fish in the fifth cohort (going backwards) have survived the following:

i) Natural mortality =

MAGE(5) for their present age this year,
+ MAGE(4) for their age last year,
+ MAGE(3) for their age the year before that,
+ .....
+ MAGE(1) for their age when recruited.

ii) Fishing mortality =

FYEAR(1)*PREC(5) for this year when they are in age class 5,
+ FYEAR(2)*PREC(4) for last year when they were in age class 4,
+ FYEAR(3)*PREC(3) for year before, when they were age class 3,
+ .....
+ FYEAR(5)*PREC(1) for the year when they were first recruited.

SUBROUTINE MORVAR(ARATE,CRATE,FYEAR,IFIRST,MAGE,MAXAG,MXSCHD,
MFLLAS,MFLSAV,MINAG,MUNLAS,MUNSAV,NAGE,NDCLAS,NSCHED,NSCHDF,NSCHDN,
YEARND,ZFMORT,ZNMORT)

Defines/edits a schedule of varying mortality rates over a range of years or age classes and transfers the schedules to rates for individual years or age classes. Fishing and natural mortality rates are handled separately. Saves a copy of the schedule on unit MUNLAS, and, optionally, on unit MUNSAV.

```
SUBROUTINE MORVED(IFIRST,MXSCHD,NS,NUMVAL,RATE,YEAR)
```

Edit mortality schedule. Edit rates and/or the years to which they apply.

```
SUBROUTINE MORVEX(IFIRST,MXSCHD,NAGE,NS,NUMVAL,RATE,YEAR)
```

Extending the rate schedule by specifying a rate and the range of years to which they apply.

```
SUBROUTINE MORVRO(ARATE,CRATE,MAXAG,MXSCHD,MFLLAS,MUNLAS,
                  NAGE,NDCLAS,NSCHED,NSCHDF,NSCHDN,YEARND,
                  ZFMORT,ZNMORT,MTYPE)
```

Opens and reads a file of varying mortality rates. Determines the number of rates, separately, for the fishing and natural mortality schedules. Displays the schedule on the terminal screen. Also defines a default range of years if a schedule has not yet been defined.

```
SUBROUTINE MORVRR(RATE,NS)
```

Interactive editing of mortality rates. Rates must be positive, and less than or equal to 3.0. If no rate is entered, then the value of RATE(NS) is unchanged.

```
SUBROUTINE MORVRY(RATE,IFIRST,NS,NUMVAL,THISYR,YEAR)
```

Interactively editing the year-ranges for the rates in a schedule of mortality rates. Subsequent schedule items can be removed by specifying a year range that exceeds the value for the subsequent item.

```
SUBROUTINE MORVSV(ARATE,CRATE,MFLSAV,MUNSAV,NDCLAS,NSCHED,NSCHDF,
                  NSCHDN,YEARND,OPTION)
```

Subroutine to save the editted schedule. If this file already exists, and OPTION is true, then the user must explicitly answer yes, when asked if the existing file's schedule is be replaced by the unit MUNSAV. If the file does not exist, one is created, and the current schedule is written into it, without asking the user.

SUBROUTINE MORVTR(ARRAY,FISHRT,MINAG,MAXAG,NAGE,NUMVAL,RATE,YEAR)

Transfers schedule arrays (RATE and YEAR values) to an array (ARRAY) for each year or age class. If FISHRT is .true., any warning messages form this routine will apply to fishing rates. If false, then natural mortality rates are assumed. For fishing rates, ARRAY is equivalent to FYEAR, RATE to ARATE, and YEAR to YEARND in the main program. For natural mortality rates, ARRAY is equivalent to MAGE, RATE to CRATE, and YEAR to NDCLAS in the main program:

MINAG	---	age of first recruitment
MAXAG	---	age of the oldest cohort
NAGE	---	number of years and cohort to simulate
NUMVAL	---	number of nonzero values in schedule.

SUBROUTINE MORZRO(ARATE,CRATE,MXSCHD,NDCLAS,YEARND)

Sets all schedule items to zero, where:

ARATE(K)	---	annual fishing mortality rate for years after YEARND(K-1) to YEARND(I). If K=1, the rate applies from the present time back to YEARND(K) years ago
CRATE(I)	---	natural mortality rate for cohort NDCLAS(I) to, but not including NDCLAS(I-1). If I=1, cohorts start from 1
MXSCHD	---	maximum number of items allowed in the schedule
NDCLAS(I)	---	oldest age class to which the natural mortality rate CRATE(I) applies
YEARND(I)	---	most "ancient" year to which the fishing mortality rate ARATE(I) applies.

SUBROUTINE MORVWR(ARATE,CRATE,IUNIT,NDCLAS,NSCHED,NSCHDF,NSCHDN,YEARND)

Writes schedule of fishing and natural mortality rates to unit IUNIT.

APPENDIX E. Subprograms in RAND.FOR

A uniform random number function to be used as an alternative to either IMSL's GGUBFS function or VAX's math library function RAN. This routine is to be linked to LFMODEL.FOR if subroutine RANNUM is to use function RAND.

FUNCTION RAND (R)

Generates a uniformly distributed series of pseudo-random numbers. Obtained from the public domain SLATEC library, it is portable among a wide variety of computers. The random numbers are generated by the linear congruential method. Given the I-th number of a pseudo-random sequence, the I+1 -st number is generated from

$$X(I+1) = (A*X(I) + C) \text{ MOD } M,$$

where M = 2**22 = 4194304, C = 1731 and several suitable values of the multiplier A are discussed in the function. Both the multiplier A and random number X are represented in double precision as two 11-bit words. The constants are chosen so that the period before the series repeats is the longest possible (4194304 values).

In order that the same numbers be generated from machine to machine, it is necessary that 23-bit integers be reducible modulo 2**11 exactly, that 23-bit integers be added exactly, and that 11-bit integers be multiplied exactly. Furthermore, if the restart option is used (where R is between 0 and 1), then the product R*2**22 = R*4194304 must be correct to the nearest integer.

Input argument, R:

if R=0., if R.LT.0.,	then the next random number of the sequence is generated, then the last generated number will be returned for possible use in a restart procedure;
if R.GT.0.,	the sequence of random numbers will start with the seed R mod 1. This seed is also returned as the value of rand provided the arithmetic is done exactly.

The value of R is left unchanged by this routine.

Output: RAND, a pseudo-random number between 0. and 1. determined from the values of the internal seeds IX1 and IX0 where: RAND = (IX1*2048 + IX0) / 4194304.

APPENDIX F. Examples of the Input File

Two examples of the input file are given below. A brief description is included in each file. See Chapter 4. for a more detailed description of the records. The basic parameters to control the simulation are read from unit INUNIT on file INFILE (Unit 10. file LFMODEL.DAT for the present version of the program).

1. Sample contents for LFMODEL.DAT when simulating silvergray rockfish:

```
S. BREVISPINIS. MALES. 3C+5A
10000                                NFISH. (TITLE is above - 80 characters)
60 9 1.00 50.0                       NAGE. MINAG. LINT. FLAG (for lineprinter plot)
57.0 0.088 -4.13                     von Bertalanffy constants: vbl. vbk. vbt
39.0500 56.9000 0.9158 1.0  9 68 Schnute constants: Y1. Y2. EXPA. B. TAU1. TAU2
0.26                                 VVB(std. dev. in mean lengths for age classes)
2.67                                 SDLAGE (std. dev. in indiv. lengths)
4.0 111111 333333                    RANGE (factr L std dev). random seeds (integer)
0.25 0.05 0.00                       ZFMORT. ZNMORT. RSIG. (part ror'tm't follows)
0.020 0.020 0.030 0.030 0.050 0.070 0.090 0.130 0.160 0.220
0.290 0.430 0.620 0.850 136*1.0
```

2. Sample contents for LFMODEL.DAT when simulating ocean perch:

```
Ocean Perch. Goose Island Females.
10000                                NFISH. (TITLE is above - 80 characters)
60 4  1.00  40.0                     NAGE. MINAG. LINT. FLAG (for lineprinter plot)
44.912 0.1366 -0.575                 von Bertalanffy constants: vbl. vbk. vbt
20.8699 44.9120 0.8723 1.0  4 83 Schnute constants: Y1. Y2. EXPA. B. TAU1. TAU2
0.26                                 VVB(std. dev. in mean lengths for age classes)
2.67                                 SDLAGE (std. dev. in indiv. lengths)
4.0 111111 333333                    RANGE (factr L std dev). random seeds (integer)
0.25 0.05 0.00                       ZFMORT. ZNMORT. RSIG. (part ror'tm't follows)
0.020 0.020 0.030 0.030 0.050 0.070 0.090 0.130 0.160 0.220
0.290 0.430 0.620 0.850 136*1.0
```

APPENDIX G. Examples of the Output Files

********** Partial output for run number 1 follows **********

Length-Frequency Distribution Simulator. (DFO. 1988)

Species simulated:

S. BREVISPINIS. MALES. 3C+5A

Specifications:

Default fishing mortality rate 0.25
Default natural mortality rate 0.05

--- USING VON BERTALANFFY'S FORMULA ---

--- USING PREVIOUS MORTALITY SCHEDULE ---

FISHING Mortality Rate Schedule:

Mortality Rate for period	Year rate ends (backward in time)
0.25	150

NATURAL Mortality Rate Schedule:

Mortality Rate for classes	Rate ends in age class:
0.05	150

Constants for Growth Models:

a) von Bertalanffy Model (Putter No. 1) :

L infinity (cm)	57.0000
T zero	-4.1300
Growth constant.K	0.0880

b) Schnute Generalized Model:

Y1. length (cm) at age. TAU1	39.0500
Y2. length (cm) at age. TAU2	56.9000
EXPA. 1st growth coefficient	0.9158
(Schnute's a=-log(EXPA)=	0.0880)
B. 2nd growth coefficient	1.0000
TAU1. age (yrs) at length Y1	9
TAU2. age (yrs) at length Y2	68

Sample size	10000
Number of age classes	60
Minimum age in simulation	9
Standard deviation for recruitment	0.00
Std. dev. in mean length for ages	0.26
Std. dev. in individual lengths	2.67
Range factor for length std. dev.	4.00

```
Initial seed. population @ age       111111
Initial seed. mean length @ age      333333
Initial seed. indiv. age-length      222223

--- Using local random number function RAND ---
```

```
Statistics for the lengths (cm) simulated:
-- ------------------------------

Length categories:
Number of categories                    42
Length of interval                   1.000
Length range - lower limit          27.000
Length range - upper limit          68.000

For fish within length range:
Mean length                         50.146
Median length (nearest 1/2 intrvl)  51.000
Standard deviation in lengths        3.999

For all fish simulated:
Number within length range           10000
Actual number that are fishable       1999
Proportion that are fishable         0.200

Simulation is complete.  Distribution follows ...

--- (Abbreviated printout - see run 4 for a complete output file) ---

Program output for this run has been written to file: LFMFILE.OU1

End of program LFMODEL
```

********** Partial output for run number 2 follows **********

Length-Frequency Distribution Simulator. (DFO. 1988)

Species simulated:

S. BREVISPINIS. MALES. 3C+5A

Specifications:

Default fishing mortality rate 0.25
Default natural mortality rate 0.05

--- USING SCHNUTE'S FORMULATION ---

--- USING PREVIOUS MORTALITY SCHEDULE

FISHING Mortality Rate Schedule:

Mortality Rate for period	Year rate ends (backward in time)
0.25	150

NATURAL Mortality Rate Schedule:

Mortality Rate for classes	Rate ends in age class:
0.05	150

Constants for Growth Models:

a) von Bertalanffy Model (Putter No. 1) :

L infinity (cm)	57.0000
T zero	-4.1300
Growth constant.K	0.0880

b) Sohnute Generalized Model:

Y1. length (cm) at age. TAU1	39.0500
Y2. length (cm) at age. TAU2	56.9000
EXPA. 1st growth coefficient	0.9158
(Sohnute's a=-log(EXPA)=	0.0880)
B. 2nd growth coefficient	1.0000
TAU1. age (yrs) at length Y1	9
TAU2. age (yrs) at length Y2	68

Sample size	10000
Number of age classes	60
Minimum age in simulation	9
Standard deviation for recruitment	0.00
Std. dev. in mean length for ages	0.26
Std. dev. in individual lengths	2.67
Range factor for length std. dev.	4.00
Initial seed. population @ age	111111
Initial seed. mean length @ age	333333
Initial seed. indiv. age-length	222223

```
--- Using local random number function RAND
```

```
Statistics for the lengths (cm) simulated:
------------------------------------------

Length categories:
Number of categories                    42
Length of interval                   1.000
Length range - lower limit          27.000
Length range - upper limit          68.000

For fish within length range:
Mean length                         50.144
Median length (nearest 1/2 intrvl)  51.000
Standard deviation in lengths        3.998

For all fish simulated:
Number within length range          10000
Actual number that are fishable      1999
Proportion that are fishable        0.200

Simulation is complete.  Distribution follows ...

--- (Abbreviated printout - see run 4 for a complete output file) ---

Program output for this run has been written to file: LFMFILE.OU2

End of program LFMODEL
```

********** Partial output for run number 3 follows **********

Length-Frequency Distribution Simulator. (DFO. 1988)

Species simulated:

S. BREVISPINIS. MALES. 3C+5A

Specifications:

Default fishing mortality rate	0.25
Default natural mortality rate	0.05

--- USING VON BERTALANFFY'S FORMULA

--- USING CONSTANT FISHING MORTALITY RATE OF 0.10 ---

--- USING CONSTANT NATURAL MORTALITY RATE OF 0.05 ---

Constants for Growth Models:

a) von Bertalanffy Model (Putter No. 1) :

L infinity (cm)	57.0000
T zero	-4.1300
Growth constant.K	0.0880

b) Sohnute Generalized Model:

Y1. length (cm) at age. TAU1	39.0500
Y2. length (cm) at age. TAU2	56.9000
EXPA. 1st growth coefficient	0.9158
(Sohnute's a=-log(EXPA)=	0.0880)
B. 2nd growth coefficient	1.0000
TAU1. age (yrs) at length Y1	9
TAU2. age (yrs) at length Y2	68

Sample size	10000
Number of age classes	60
Minimum age in simulation	9
Standard deviation for recruitment	0.00
Std. dev. in mean length for ages	0.26
Std. dev. in individual lengths	2.67
Range factor for length std. dev.	4.00
Initial seed. population @ age	111111
Initial seed. mean length @ age	333333
Initial seed. indiv. age-length	222223

--- Using local random number function RAND ---

```
Statistics for the lengths (cm) simulated:
------------------------------------------

Length categories:
Number of categories                    42
Length of interval                   1.000
Length range - lower limit          27.000
Length range - upper limit          68.000

For fish within length range:
Mean length                         51.715
Median length (nearest 1/2 intrvl)  52.500
Standard deviation in lengths        3.945

For all fish simulated:
Number within length range          10000
Actual number that are fishable      3236
Proportion that are fishable        0.324

Simulation is complete.  Distribution follows ...

--- (Abbreviated printout - see run 4 for a complete output file) ---

Program output for this run has been written to file: LFMFILE.OU3

End of program LFMODEL
```

********** Complete output for run number 4 follows **********

Length-Frequency Distribution Simulator. (DFO. 1988)

Species simulated:

S. BREVISPINIS. MALES. 3C+5A

Specifications:

Default fishing mortality rate 0.25
Default natural mortality rate 0.05

--- USING SCHNUTE'S FORMULATION ---

--- USING VARIABLE MORTALITY ---

Mortality Rate Schedule in the Simulation:

FISHING Mortality Rate Schedule:

Mortality Rate for period	Year rate ends (backward in time)
0.25	10
0.00	15
0.10	150

NATURAL Mortality Rate Schedule:

Mortality Rate for classes	Rate ends in age class:
0.10	10
0.02	40

*** Specified NATURAL mortality schedule was too short.
Rate of 0.02 assumed for fish from 49 to 68 years old.

Constants for Growth Models:

a) von Bertalanffy Model (Putter No. 1) :
L infinity (cm) 57.0000
T zero -4.1300
Growth constant.K 0.0880

b) Schnute Generalized Model:
Y1. length (cm) at age. TAU1 39.0500
Y2. length (cm) at age. TAU2 56.9000
EXPA. 1st growth coefficient 0.9158
(Schnute's a=-log(EXPA)= 0.0880)
B. 2nd growth coefficient 1.0000
TAU1. age (yrs) at length Y1 9
TAU2. age (yrs) at length Y2 68

Sample size 10000
Number of age classes 60
Minimum age in simulation 9

Standard deviation for recruitment	0.00
Std. dev. in mean length for ages	0.26
Std. dev. in individual lengths	2.67
Range factor for length std. dev.	4.00
Initial seed. population @ age	111111
Initial seed. mean length @ age	333333
Initial seed. indiv. age-length	222223

--- Using local random number function RAND ---

Statistics for the lengths (cm) simulated:

Length categories:	
Number of categories	42
Length of interval	1.000
Length range - lower limit	27.000
Length range - upper limit	68.000
For fish within length range:	
Mean length	50.931
Median length (nearest 1/2 intrvl)	51.500
Standard deviation in lengths	4.389
For all fish simulated:	
Number within length range	10000
Actual number that are fishable	2129
Proportion that are fishable	0.213

Simulation is complete. Distribution follows ...

Age	Mean Length	Act'l Abund. Before Recr.	% at Age of Fishable Pop.	Part Recr. Factor	Z. Accum'ltd Mort. Rate	EXP(-Z)	Natural Rate	Fishing Years ago	Fishing Rate
	AGE DISTRIBUTION OF FISH IN SAMPLE POPULATION						MORTALITY RATE COMPONENTS		
9	38.47	12.24	1.15	0.02	0.11	0.900	0.10	0	0.25
10	40.20	11.02	1.04	0.02	0.21	0.811	0.10	1	0.25
11	41.44	9.90	1.39	0.03	0.32	0.728	0.10	2	0.25
12	42.62	8.89	1.25	0.03	0.42	0.654	0.10	3	0.25
13	44.56	7.95	1.87	0.05	0.54	0.584	0.10	4	0.25
14	45.58	7.06	2.32	0.07	0.65	0.519	0.10	5	0.25
15	46.59	6.25	2.64	0.09	0.78	0.460	0.10	6	0.25
16	47.69	5.47	3.34	0.13	0.91	0.403	0.10	7	0.25
17	48.30	4.76	3.58	0.16	1.05	0.350	0.10	8	0.25
18	48.54	4.08	4.21	0.22	1.21	0.300	0.10	9	0.25
19	49.85	3.73	5.09	0.29	1.29	0.275	0.02	10	0.00
20	50.24	3.30	6.67	0.43	1.41	0.243	0.02	11	0.00
21	50.69	2.79	8.14	0.62	1.58	0.205	0.02	12	0.00
22	51.05	2.23	8.91	0.85	1.81	0.164	0.02	13	0.00
23	52.13	1.72	8.10	1.00	2.06	0.127	0.02	14	0.00
24	52.06	1.34	6.28	1.00	2.32	0.098	0.02	15	0.10
25	52.36	1.04	4.89	1.00	2.57	0.077	0.02	16	0.10
26	52.67	0.82	3.85	1.00	2.81	0.060	0.02	17	0.10
27	53.24	0.65	3.05	1.00	3.04	0.048	0.02	18	0.10
28	53.56	0.52	2.45	1.00	3.26	0.038	0.02	19	0.10
29	53.63	0.42	1.99	1.00	3.47	0.031	0.02	20	0.10
30	53.79	0.36	1.68	1.00	3.64	0.026	0.02	21	0.10
31	54.22	0.31	1.48	1.00	3.77	0.023	0.02	22	0.10
32	54.68	0.29	1.37	1.00	3.84	0.021	0.02	23	0.10
33	54.97	0.28	1.32	1.00	3.88	0.021	0.02	24	0.10
34	54.92	0.27	1.25	1.00	3.93	0.020	0.02	25	0.10
35	55.27	0.25	1.18	1.00	3.99	0.018	0.02	26	0.10
36	54.97	0.23	1.08	1.00	4.08	0.017	0.02	27	0.10
37	55.85	0.21	0.98	1.00	4.18	0.015	0.02	28	0.10
38	55.60	0.18	0.87	1.00	4.30	0.014	0.02	29	0.10
39	55.76	0.16	0.77	1.00	4.42	0.012	0.02	30	0.10
40	55.92	0.15	0.68	1.00	4.54	0.011	0.02	31	0.10
41	56.00	0.13	0.60	1.00	4.66	0.009	0.02	32	0.10
42	55.79	0.11	0.54	1.00	4.78	0.008	0.02	33	0.10
43	56.51	0.10	0.48	1.00	4.90	0.007	0.02	34	0.10
44	56.05	0.09	0.42	1.00	5.02	0.007	0.02	35	0.10
45	56.17	0.08	0.37	1.00	5.14	0.006	0.02	36	0.10
46	56.18	0.07	0.33	1.00	5.26	0.005	0.02	37	0.10
47	56.91	0.06	0.29	1.00	5.38	0.005	0.02	38	0.10
48	55.95	0.06	0.26	1.00	5.50	0.004	0.02	39	0.10
49	56.43	0.05	0.23	1.00	5.62	0.004	0.02	40	0.10
50	56.43	0.04	0.21	1.00	5.74	0.003	0.02	41	0.10
51	56.06	0.04	0.18	1.00	5.86	0.003	0.02	42	0.10
52	56.56	0.03	0.16	1.00	5.98	0.003	0.02	43	0.10
53	56.59	0.03	0.14	1.00	6.10	0.002	0.02	44	0.10

54	56.83	0.03	0.13	1.00	6.22	0.002	0.02	45	0.10
55	56.68	0.02	0.11	1.00	6.34	0.002	0.02	46	0.10
56	56.76	0.02	0.10	1.00	6.46	0.002	0.02	47	0.10
57	56.82	0.02	0.09	1.00	6.58	0.001	0.02	48	0.10
58	57.29	0.02	0.08	1.00	6.70	0.001	0.02	49	0.10
59	56.79	0.01	0.07	1.00	6.82	0.001	0.02	50	0.10
60	56.34	0.01	0.06	1.00	6.94	0.001	0.02	51	0.10
61	56.72	0.01	0.05	1.00	7.06	0.001	0.02	52	0.10
62	56.96	0.01	0.05	1.00	7.18	0.001	0.02	53	0.10
63	56.42	0.01	0.04	1.00	7.30	0.001	0.02	54	0.10
64	56.74	0.01	0.04	1.00	7.42	0.001	0.02	55	0.10
65	56.76	0.01	0.03	1.00	7.54	0.001	0.02	56	0.10
66	56.95	0.01	0.03	1.00	7.66	0.000	0.02	57	0.10
67	56.71	0.01	0.03	1.00	7.78	0.000	0.02	58	0.10
68	56.64	0.01	0.02	1.00	7.90	0.000	0.02	59	0.10

DISTRIBUTION OF FISH IN SAMPLE POPULATION

Freq'noy	Prop of Distrib	Length (cm) +/- 0.5	Frequenoy
0	.000	27	
0	.000	28	
0	.000	29	
0	.000	30	
1	.000	31	
0	.000	32	
5	.001	33	
6	.001	34	
16	.002	35	
17	.002	36	
32	.003	37	
35	.004	38	
59	.006	39	
68	.007	40	
79	.008	41	
111	.011	42	
159	.016	43	
186	.019	44	
249	.025	45	
365	.036	46	
497	.050	47	
625	.063	48	
826	.083	49	
900	.090	50	
1050	.105	51	
999	.100	52	
922	.092	53	
818	.082	54	
636	.064	55	
479	.048	56	
334	.033	57	
239	.024	58	
140	.014	59	
79	.008	60	
38	.004	61	
20	.002	62	
9	.001	63	
0	.000	64	
1	.000	65	
0	.000	66	
0	.000	67	
0	.000	68	

```
DATA SET HEADINGS IN DISTRIBUTION DATA SUMMARY FILE: LFMFILE.PRO
----------------------------------------------------------------


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU1
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  1
-----------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU2
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  2
-----------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU3
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  3
-----------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


Length-frequency distribution is being added to file: LFMFILE.PRO
with the data set heading as below:

LFMODEL.FOR summary for ..... Variable Mortality
```

```
For details see file ..... LFMFILE.OU4
No. of values after header ... 42 at  1.0 cm intervals
To read above values ..... format(T33,I2,T39,F4.1)
To read data use ..... format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  4
-------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


Program output for this run has been written to file: LFMFILE.OU4

End of program LFMODEL
```

********** Partial output for run number 5 follows **********

Length-Frequency Distribution Simulator. (DFO. 1988)

Species simulated:

S. BREVISPINIS. MALES. 3C+5A

Specifications:

Default fishing mortality rate 0.25
Default natural mortality rate 0.05

--- USING VON BERTALANFFY'S FORMULA ---

--- USING VARIABLE MORTALITY ---

Mortality Rate Schedule in the Simulation:

FISHING Mortality Rate Schedule:

Mortality Rate for period	Year rate ends (backward in time)
0.25	150

NATURAL Mortality Rate Schedule:

Mortality Rate for classes	Rate ends in age class:
0.05	150

Constants for Growth Models:

a) von Bertalanffy Model (Putter No. 1) :

L infinity (cm)	57.0000
T zero	-4.1300
Growth constant.K	0.0880

b) Sohnute Generalized Model:

Y1. length (cm) at age. TAU1	39.0500
Y2. length (cm) at age. TAU2	56.9000
EXPA. 1st growth coefficient	0.9158
(Sohnute's a=-log(EXPA)=	0.0880)
B. 2nd growth coefficient	1.0000
TAU1. age (yrs) at length Y1	9
TAU2. age (yrs) at length Y2	68

Sample size	10000
Number of age classes	60
Minimum age in simulation	9
Standard deviation for recruitment	0.00
Std. dev. in mean length for ages	0.26
Std. dev. in individual lengths	2.67
Range factor for length std. dev.	4.00
Initial seed. population @ age	111111

```
Initial seed. mean length @ age      333333
Initial seed. indiv. age-length      222223

--- Using local random number function RAND ---
```

```
 Statistics for the lengths (cm) simulated:
 ------------------------------------------

 Length categories:
 Number of categories                    42
 Length of interval                   1.000
 Length range - lower limit          27.000
 Length range - upper limit          68.000

 For fish within length range:
 Mean length                         50.146
 Median length (nearest 1/2 intrvl)  51.000
 Standard deviation in lengths        3.999

 For all fish simulated:
 Number within length range           10000
 Actual number that are fishable       1999
 Proportion that are fishable         0.200

 Simulation is complete.  Distribution follows ...

--- (Abbreviated printout - see run 4 for a complete output file) ---

**************************************************************

 WARNING:- The maximum number of data sets on file LFMFILE.PRO
 has been reached.  Old data sets will be lost now or with the
 next run of this program.
 Please remove any unwanted data sets and/or rename the file.
 Next run will overwrite the  5-th data only (i.e. Run no.  5)

**************************************************************


 Length-frequency distribution is being added to file: LFMFILE.PRO
 with the data set heading as below:

 LFMODEL.FOR summary for ..... Variable Mortality
 For details see file ........ LFMFILE.OU5
 No. of values after header ... 42 at  1.0 cm intervals
 To read above values ........ format(T33,I2,T39,F4.1)
 To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

 Distribution of Fish in Sample Population for Run Number  5
 -----------------------------------------------------------

 Run      Length          Frequency
 No.    +/- 0.5 cm
```

```
**************************************************************

 WARNING:- Output from this run. written to file: LFMFILE.OU5
           All output file names have now been used.

           Please delete or rename some of the following
           files before rerunning this program.

           Note:- The program will use the first unused
           file name from this list.

 Order of Search         File Name
        1                LFMFILE.OU1
        2                LFMFILE.OU2
        3                LFMFILE.OU3
        4                LFMFILE.OU4
        5                LFMFILE.OU5

**************************************************************
```

```
DATA SET HEADINGS IN DISTRIBUTION DATA SUMMARY FILE: LFMFILE.PRO
----------------------------------------------------------------


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU1
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  1
-----------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU2
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  2
-----------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU3
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  3
-----------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Variable Mortality
For details see file ........ LFMFILE.OU4
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)
```

```
Distribution of Fish in Sample Population for Run Number  4
-------------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


DATA SET HEADINGS IN DISTRIBUTION DATA SUMMARY FILE: LFMFILE.PRO
---------------------------------------------------------------

LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU1
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  1
-------------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU2
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  2
-------------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Constant Mortality
For details see file ........ LFMFILE.OU3
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33,I2,T39,F4.1)
To read data use ............ format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number  3
-------------------------------------------------------------
```

```
Run      Length          Frequency
No.    +/- 0.5 cm


LFMODEL.FOR summary for ..... Variable Mortality
For details see file ........ LFMFILE.OU4
No. of values after header ... 42 at  1.0 cm intervals
To read above values ........ format(T33.I2.T39.F4.1)
To read data use ............ format(T2.I2.T5.F10.1.T20.I12)

Distribution of Fish in Sample Population for Run Number  4
------------------------------------------------------------

Run      Length          Frequency
No.    +/- 0.5 cm
```

********** The contents of LFMFILE.PRO follows **********

LFMODEL.FOR summary for ..... Constant Mortality
For details see file LFMFILE.OU1
No. of values after header ... 42 at 1.0 cm intervals
To read above values ..... format(T33,I2,T39,F4.1)
To read data use ..... format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number 1

Run No.	Length +/- 0.5 cm	Frequency
1	27.0	0
1	28.0	0
1	29.0	0
1	30.0	0
1	31.0	1
1	32.0	0
1	33.0	4
1	34.0	6
1	35.0	16
1	36.0	16
1	37.0	26
1	38.0	33
1	39.0	55
1	40.0	66
1	41.0	89
1	42.0	113
1	43.0	184
1	44.0	217
1	45.0	292
1	46.0	464
1	47.0	626
1	48.0	778
1	49.0	967
1	50.0	1096
1	51.0	1131
1	52.0	1014
1	53.0	886
1	54.0	730
1	55.0	479
1	56.0	317
1	57.0	210
1	58.0	96
1	59.0	47
1	60.0	31
1	61.0	7
1	62.0	3
1	63.0	0

LFMODEL.FOR summary for ..... Constant Mortality
For details see file LFMFILE.OU2
No. of values after header ... 42 at 1.0 cm intervals
To read above values ..... format(T33.I2.T39.F4.1)
To read data use ..... format(T2.I2.T5.F10.1.T20.I12)

Distribution of Fish in Sample Population for Run Number 2

Run No.	Length +/- 0.5 cm	Frequency
2	27.0	0
2	28.0	0
2	29.0	0
2	30.0	0
2	31.0	1
2	32.0	0
2	33.0	4
2	34.0	6
2	35.0	16
2	36.0	17
2	37.0	25
2	38.0	34
2	39.0	54
2	40.0	66
2	41.0	91
2	42.0	112
2	43.0	184
2	44.0	217
2	45.0	293
2	46.0	463
2	47.0	627
2	48.0	779
2	49.0	968
2	50.0	1096
2	51.0	1129
2	52.0	1016
2	53.0	882
2	54.0	731
2	55.0	480
2	56.0	315
2	57.0	210
2	58.0	96
2	59.0	47
2	60.0	31
2	61.0	7
2	62.0	3
2	63.0	0
2	64.0	0
2	65.0	0

000

66.0
67.0
68.0

222

LFMODEL.FOR summary for ..... Constant Mortality
For details see file LFMFILE.OU3
No. of values after header ... 42 at 1.0 cm intervals
To read above values ..... format(T33,I2,T39,F4.1)
To read data use ..... format(T2,I2,T5,F10.1,T20,I12)

Distribution of Fish in Sample Population for Run Number 3

Run No.	Length +/- 0.5 cm	Frequency
3	27.0	0
3	28.0	0
3	29.0	0
3	30.0	0
3	31.0	0
3	32.0	0
3	33.0	2
3	34.0	3
3	35.0	4
3	36.0	14
3	37.0	16
3	38.0	16
3	39.0	34
3	40.0	34
3	41.0	33
3	42.0	63
3	43.0	78
3	44.0	123
3	45.0	185
3	46.0	261
3	47.0	388
3	48.0	571
3	49.0	709
3	50.0	907
3	51.0	1086
3	52.0	1140
3	53.0	1031
3	54.0	999
3	55.0	741
3	56.0	579
3	57.0	412
3	58.0	265
3	59.0	149
3	60.0	83
3	61.0	45
3	62.0	20
3	63.0	8
3	64.0	0
3	65.0	1

APPENDIX H. Examples of Terminal Display for an Interactive Session

********** Run with all defaults. follows **********

```
$ run lfmodel

Length-Frequency Distribution Simulator.  (DFO. 1988)
-----------------------------------------------------

Program output for this run will be written to file: LFMFILE.OUT


Do you wish to use Schnute's Growth Formulation?  [y/n] (default=n)
(If default or n. then will use von Bertalanffy's)

--- USING VON BERTALANFFY'S FORMULA ---

Do you wish to use the mortality rates from the last run? [y/n] (default=y):
If not. then variable rates can be displayed &  optionally  editted.

--- USING PREVIOUS MORTALITY SCHEDULE ---

FISHING Mortality Rate Schedule:        NATURAL Mortality Rate Schedule:

Mortality Rate    Year rate ends        Mortality Rate      Rate ends in
  for period    (backward in time)        for classes       age class:
     0.25             150                    0.05             150

Press RETURN to continue ...


INTERACTIVE INPUT ENDED.

Now simulating the length distribution.  Please wait ...


Simulation is complete.  Distribution follows ...

                              DISTRIBUTION OF FISH IN SAMPLE POPULATION
                              -----------------------------------------

Freq'ncy  Prop of  Length (cm)                    Frequency
          Distrib   +/- 0.5

    0      .000        27     *
```

```
     0      .000        28    *
     0      .000        29    *
     0      .000        30    *
     1      .000        31    *
     0      .000        32    *
     4      .000        33    *
     6      .001        34    *
    16      .002        35    |*
    16      .002        36    |*
    26      .003        37    |*
    33      .003        38    |*
    55      .005        39    | *
    66      .007        40    |  *
    89      .009        41    |   *
   113      .011        42    |    *
   184      .018        43    |     *
   217      .022        44    |      *
   292      .029        45    |       *
   464      .046        46    |         *
   626      .063        47    |          *
   778      .078        48    |           *
   967      .097        49    |            *
  1096      .110        50    |-------------------------------------*--
  1131      .113        51    |                                      *
  1014      .101        52    |                                   *
   886      .089        53    |                                *
   730      .073        54    |                             *
   479      .048        55    |                          *
   317      .032        56    |                       *
   210      .021        57    |                    *
    98      .010        58    |                 *
    47      .005        59    |               *
    31      .003        60    |*
     7      .001        61    *
     3      .000        62    *
     0      .000        63    *
     0      .000        64    *
     0      .000        65    *
     0      .000        66    *
     0      .000        67    *
     0      .000        68    *
```

Length-frequency distribution is being added to file: LFMFILE.PRO

Press RETURN to continue ...

Program output for this run has been written to file: LFMFILE.OU1

End of program LFMODEL

```
************** Run with all defaults except for Schnute equation ***************
$ run lfmodel

Length-Frequency Distribution Simulator.  (DFO. 1988)
-------------------------------------------------------

Program output for this run will be written to file: LFMFILE.OU2


Do you wish to use Schnute's Growth Formulation?  [y/n] (default-n)
(If default or n. then will use von Bertalanffy's)
y

--- USING SCHNUTE'S FORMULATION ---

Do you wish to use the mortality rates from the last run? [y/n] (default-y):
If not. then variable rates can be displayed &  optionally  editted.
y

--- USING PREVIOUS MORTALITY SCHEDULE ---

FISHING Mortality Rate Schedule:          NATURAL Mortality Rate Schedule:

Mortality Rate     Year rate ends         Mortality Rate      Rate ends in
  for period     (backward in time)          for classes       age class:
     0.25              150                      0.05              150

Press RETURN to continue ...


INTERACTIVE INPUT ENDED.

Now simulating the length distribution.  Please wait ...


Simulation is complete.  Distribution follows ...

--- (Distribution has been removed for display purposes only) ---

Length-frequency distribution is being added to file: LFMFILE.PRO

Press RETURN to continue ...

Program output for this run has been written to file: LFMFILE.OU2

End of program LFMODEL
```

```
************** Run with constant mortality follows ******************************
$ run lfmodel

Length-Frequency Distribution Simulator.  (DFO. 1988)
-----------------------------------------------------

Program output for this run will be written to file: LFMFILE.OU3


Do you wish to use Schnute's Growth Formulation?  [y/n] (default=n)
(If default or n. then will use von Bertalanffy's)


--- USING VON BERTALANFFY'S FORMULA ---

Do you wish to use the mortality rates from the last run? [y/n] (default=y):
If not. then variable rates can be displayed &  optionally  editted.
n

Do you wish to use a constant mortality rate? [y/n] (default=y):
y

--- USING CONSTANT MORTALITY ---


Enter constant FISHING mortality rate: (default =  0.25 )
(1 to 9 digit-decimal number from 0.0 to 3.0 (format = F10.0))
.10

--- USING CONSTANT FISHING MORTALITY RATE OF  0.10 ---

Enter constant NATURAL mortality rate: (default =  0.05 )
(1 to 9 digit-decimal number from 0.0 to 3.0 (format = F10.0))


--- USING CONSTANT NATURAL MORTALITY RATE OF  0.05 ---

The current constant rate schedule has been saved on file LFMORTRT.LAS


INTERACTIVE INPUT ENDED.

Now simulating the length distribution.  Please wait ...



Simulation is complete.  Distribution follows ...
```

```
--- (Distribution has been removed for display purposes only) ---
Length frequency distribution is being added to file: LFMFILE.PRO
Press RETURN to continue ...

Program output for this run has been written to file: LFMFILE.OU3
End of program LFMODEL
```

```
************** Run with variable mortality and editting. follows ****************
$ run lfmodel

Length-Frequency Distribution Simulator.  (DFO. 1988)
-------------------------- ---                  --------

Program output for this run will be written to file: LFMFILE.OU4


Do you wish to use Schnute's Growth Formulation?  [y/n] (default-n)
(If default or n. then will use von Bertalanffy's)
y

--- USING SCHNUTE'S FORMULATION ---

Do you wish to use the mortality rates from the last run? [y/n] (default-y):
If not. then variable rates can be displayed &  optionally  editted.
n

Do you wish to use a constant mortality rate? [y/n] (default-y):
n

--- USING VARIABLE MORTALITY ---

DEFAULT Mortality Schedule:

FISHING Mortality Rate Schedule:        NATURAL Mortality Rate Schedule:

Mortality Rate     Year rate ends       Mortality Rate      Rate ends in
  for period     (backward in time)       for classes       age class:
     0.10              60                    0.05              60

Do you wish to use the above schedule? [y/n] (default-y):
n

Do you wish to see the schedule saved from a prior run?  [y/n] (default-n)
(If default or n. then will edit DEFAULT schedule)
y

EXPLICITLY-SAVED Mortality Schedule:

FISHING Mortality Rate Schedule:        NATURAL Mortality Rate Schedule:

Mortality Rate     Year rate ends       Mortality Rate      Rate ends in
  for period     (backward in time)       for classes       age class:
     0.25             150                    0.05             150

Do you wish to use the above schedule? [y/n] (default-y):
(If n. then will edit EXPLICITLY-SAVED schedule)
n
```

```
EDITTING EXPLICITLY SAVED MORTALITY SCHEDULE:


Do you wish to edit the FISHING mortalities? [y/n] (default-n):
y

Schedule item   1 :
    Rate of  0.25 will apply from years/age classes   1 to 150 inclusive.

    Is this okay? [y/n] (default-y):
n
    Present rate is  0.25.  Enter new rate? : (default: no change)
    (1 to 9 digit-decimal number from 0.0 to 3.0 (format - F10.0))

    Rate remains at  0.25.

    It will apply from years age-classes   1 to 150 inclusive.

    Change last year/age-class (150)?  Enter new value: (default: no change)
10

Revised item    1)
    Rate of  0.25 will apply from years/age-classes   1 to  10 inclusive.


Do you wish to EXTEND this schedule? [y/n] (default-n):
y

EXTENDING FISHING MORTALITY SCHEDULE:

Schedule item   2 :
    Rate of  0.25 will apply from years/age classes  11 to  60 inclusive.

    Is this okay? [y/n] (default-y):
n
    Present rate is  0.25.  Enter new rate? : (default: no change)
    (1 to 9 digit-decimal number from 0.0 to 3.0 (format - F10.0))
0
    Previous rate of  0.25 has been replaced by  0.00.

    It will apply from years/age-classes  11 to  60 inclusive.

    Change last year/age-class ( 60)?  Enter new value: (default: no change)
15

Revised item    2)
    Rate of  0.00 will apply from years/age-classes  11 to  15 inclusive.
```

```
Do you wish to EXTEND this schedule? [y/n] (default-n):
y

EXTENDING FISHING MORTALITY SCHEDULE:


Schedule item   3 :
    Rate of  0.00 will apply from years/age-classes  16 to  60 inclusive.

    Is this okay? [y/n] (default-y):
n
    Present rate is  0.00.  Enter new rate? : (default: no change)
    (1 to 9 digit-decimal number from 0.0 to 3.0 (format - F10.0))
.10
    Previous rate of  0.00 has been replaced by  0.10.

    It will apply from years/age classes  16 to  60 inclusive.

    Change last year/age-class ( 60)?  Enter new value: (default: no change)
150

Revised item    3)
    Rate of  0.10 will apply from years/age-classes  16 to 150 inclusive.


Do you wish to EXTEND this schedule? [y/n] (default-n):


Do you wish to edit the NATURAL mortalities? [y/n] (default-n):
y

Schedule item   1 :
    Rate of  0.05 will apply from years/age-classes   1 to 150 inclusive.

    Is this okay? [y/n] (default-y):
n
    Present rate is  0.05.  Enter new rate? : (default: no change)
    (1 to 9 digit-decimal number from 0.0 to 3.0 (format - F10.0))
.1
    Previous rate of  0.05 has been replaced by  0.10.

    It will apply from years/age-classes   1 to 150 inclusive.

    Change last year/age-class (150)?  Enter new value: (default: no change)
10
```

```
Revised item    1)
    Rate of  0.10 will apply from years/age-classes   1 to  10 inclusive.


Do you wish to EXTEND this schedule? [y/n] (default-n):
y

EXTENDING NATURAL MORTALITY SCHEDULE:


Schedule item   2 :
    Rate of  0.10 will apply from years/age-classes  11 to  60 inclusive.

    Is this okay? [y/n] (default-y):
n
    Present rate is  0.10.  Enter new rate? : (default: no change)
    (1 to 9 digit-decimal number from 0.0 to 3.0 (format - F10.0))
34
    Rate entered (  34.0000    ) is invalid.  Please reenter:
    Present rate is  0.10.  Enter new rate? : (default: no change)
    (1 to 9 digit-decimal number from 0.0 to 3.0 (format - F10.0))
.02
    Previous rate of  0.10 has been replaced by  0.02.

    It will apply from years/age-classes  11 to  60 inclusive.

    Change last year/age-class ( 60)?  Enter new value: (default: no change)
5
    Value must be .ge.  11. Please reenter:
40

Revised item    2)
    Rate of  0.02 will apply from years/age-classes  11 to  40 inclusive.


Do you wish to EXTEND this schedule? [y/n] (default-n):


FINAL Mortality Schedule:

FISHING Mortality Rate Schedule:        NATURAL Mortality Rate Schedule:

Mortality Rate    Year rate ends        Mortality Rate     Rate ends in
  for period    (backward in time)        for classes      age class:
     0.25              10                    0.10              10
     0.00              15                    0.02              40
     0.10             150

*** Specified NATURAL mortality schedule was too short.
```

```
Rate of  0.02 assumed for fish from  49 to  68 years old.

The latest rate schedule (  3 item(s)) has been saved on file LFMORTRT.LAS


**** An EXPLICITLY-SAVED rate schedule already exists. ****
Do you wish to replace it with the above schedule? [y/n] (default-n):


INTERACTIVE INPUT ENDED.
Now simulating the length distribution.  Please wait ...


Simulation is complete.  Distribution follows ...
--- (Distribution has been removed for display purposes only) ---
Length-frequency distribution is being added to file: LFMFILE.PRO
Press RETURN to continue ...

Program output for this run has been written to file: LFMFILE.OU4
End of program LFMODEL
```

```
************** Run with output files about to be filled. follows ****************
$ run lfmodel

Length-Frequency Distribution Simulator.  (DFO. 1988)
-----------------------------------------------------

Program output for this run will be written to file: LFMFILE.OUS


Do you wish to use Schnute's Growth Formulation?  [y/n] (default-n)
(If default or n. then will use von Bertalanffy's)

--- USING VON BERTALANFFY'S FORMULA ---

Do you wish to use the mortality rates from the last run? [y/n] (default-y):
If not. then variable rates can be displayed &  optionally  editted.
n

Do you wish to use a constant mortality rate? [y/n] (default-y):
n

--- USING VARIABLE MORTALITY ---

DEFAULT Mortality Schedule:

FISHING Mortality Rate Schedule:          NATURAL Mortality Rate Schedule:

Mortality Rate     Year rate ends         Mortality Rate      Rate ends in
  for period     (backward in time)          for classes       age class:
      0.25              10                      0.10              10
      0.00              15                      0.02              40
      0.10             150

Do you wish to use the above schedule? [y/n] (default-y):
n

Do you wish to see the schedule saved from a prior run?  [y/n] (default-n)
(If default or n. then will edit DEFAULT schedule)
y

EXPLICITLY-SAVED Mortality Schedule:

FISHING Mortality Rate Schedule:          NATURAL Mortality Rate Schedule:

Mortality Rate     Year rate ends         Mortality Rate      Rate ends in
  for period     (backward in time)          for classes       age class:
      0.25             150                      0.05             150

Do you wish to use the above schedule? [y/n] (default-y):
```

```
(If n. then will edit EXPLICITLY SAVED schedule)

FINAL Mortality Schedule:

FISHING Mortality Rate Schedule:          NATURAL Mortality Rate Schedule:

Mortality Rate    Year rate ends          Mortality Rate      Rate ends in
  for period    (backward in time)          for classes       age class:
     0.25             150                      0.05              150

The latest rate schedule (  1 item(s)) has been saved on file LFMORTRT.LAS


**** An EXPLICITLY-SAVED rate schedule already exists. ****
Do you wish to replace it with the above schedule? [y/n] (default=n):
y

The latest rate schedule (  1 item(s)) has been saved on file LFMORTRT.SAV


INTERACTIVE INPUT ENDED.

Now simulating the length distribution.  Please wait ...


Simulation is complete.  Distribution follows ...

--- (Distribution has been removed for display purposes only) ---

************************************************************

WARNING:- The maximum number of data sets on file LFMFILE.PRO
has been reached.  Old data sets will be lost now or with the
next run of this program.
Please remove any unwanted data sets and/or rename the file.
Next run will overwrite the  5-th data only (i.e. Run no.  5)

************************************************************

Length-frequency distribution is being added to file: LFMFILE.PRO

Press RETURN to continue ...
```

```
**************************************************************

WARNING:- Output from this run. written to file: LFMFILE.OU5
          All output file names have now been used.

          Please delete or rename some of the following
          files before rerunning this program.

          Note:- The program will use the first unused
          file name from this list.

Order of Search          File Name
       1                 LFMFILE.OU1
       2                 LFMFILE.OU2
       3                 LFMFILE.OU3
       4                 LFMFILE.OU4
       5                 LFMFILE.OU5

**************************************************************
```

```
************** Run with no output files available, follows **********************
$ run lfmodel

Length Frequency Distribution Simulator.  (DFO, 1988)
------ --------- ------------ ----------  ---- -----

************* Program Run Cancelled *************
  Creating a new output file for this program
  will overwrite output from previous runs.
  Please delete or rename some of the following
  files before rerunning this program.

  Note:- The program will use the first unused
  file name from this list.

Order of Search          File Name
      1                  LFMFILE.OU1
      2                  LFMFILE.OU2
      3                  LFMFILE.OU3
      4                  LFMFILE.OU4
      5                  LFMFILE.OU5

Program cancelled
-----------------------------------------------------------------
```

APPENDIX I. Program Listing for File LFMODEL.FOR

The mainprogram and the length freqency simulator routines.

```
C********** LFMODEL.FOR ***************************************************
C                                                                           *
C     Simulator for generating observed length frequencies of fish          *
C     with either von Bertalanffy growth characteristics (Putter No. 1)*
C     or Schnute generalized growth characteristics.  Mortality rates,      *
C     from fishing and natural causes, can be specified for specific        *
C     years and age classes.                                                *
C                                                                           *
C***************************************************************************
C
C  History of Versions of this program:
C
C  1988 version by M.C. Rasmussen. (Last revision March 1988).
C  For D.F.O., Canada, Pacific Biological Station, Nanaimo, B.C.
C
C  Features:
C
C  1)  Function RANNUM added - a dummy function to allow the use of
C      alternate pseudo-random number functions, depending upon the
C      operating system, fortran compiler, and libraries available.
C      A function RAND has been added to this program, to generate
C      pseudo-random numbers when neither the IMSL library function
C      GGUBFS nor the VAX math library function RAN is available.
C      RAND was obtained from the public domain SLATEC library.
C
C  2)  Length distribution determined using a specified length interval.
C      The number of frequencies in the distribution is determined from
C      this interval and the expected length minima and maxima.  Output
C      is only over this range.
C
C  3)  Subroutine GETMS - inner DO-loop removed and calculation of
C      S(I) varying with age was removed, and replaced by a constant
C      term SDLAGE, from the input file (std. dev. in calculating
C      individual length).  The term VBB, std. dev. in mean length for
C      age classes was implemented directly.  (The original program
C      truncated VBB after the first decimal digit (eg. 0.26 became
C      0.2)).
C
C  4)  Removed all references to the following variables:
C      PASIM, JMIN, JMAX, FREQA, RF, FAL, SFAL, SFREQA, JSMIN, JSMAX,
C      INDEX, INDEXS, PA, RTOT, KSWICH, ISLEN, PASUM, GAM, LDR, IARR,
C      PSUM, PSIM, SEED, FMORT as they were no longer necessary for the
C      current form of the program.
C
C  5)  Common block (OVBLK) reorganized and dimensions for the maximum
C      number of frequencies and years specified by parameter named
C      constants throughout the program.  Some variables were removed
C      from the common block as they were no longer needed outside of
C      their originating routine.
C
C  6)  Subroutine GETPS was modified.  Expressions for POPSUM and P
C      now use array elements SAMORT(J) in place of J*ZMORT.  This
C      allows each age class to have its own mortality rate.  Natural
C      mortality and fishing mortality are now considered individually,
C      along with the partial recruitment factor, when determining
C      SAMORT, the effective mortality rate experienced by an age class.
C
C  7)  Module MORATE was written to read &/or edit a mortality
C      rate schedule to be used optionally in place of ZMORT.  Uses
C      (or creates) a file called LFMORTRT.LAS on unit 11.  The
C      may also save the editted schedule for use in a subsequent run.
C      This schedule is saved in a file called LFMORTRT.SAV
C
C      If constant mortality is to be used, the user may specify a rate
C      other than ZMORT (now called ZFMORT for fishing, ZNMORT for
C      natural causes).
C
C      The mortality rates for each year from the present back NAGE
C      years are summed to give SAMORT(J), where J=1 to NAGE.
C      This is the equivalent of J*ZFMORT or J*ZNMORT when mortality
C      is constant.
C
```

```
C     Now handles separate mortality schedules for fishing and natural
C     mortalities.
C
C  8) Common block LGUNIT added to define logical units for I/O for
C     interactive use and for the output of the results.
C
C  9) In NASIM, an used random number function call was removed. This
C     occurred between the calls to GETPS and GETMS. This effectively
C     started GETMS at 1 number earlier in the series of random numbers
C     used to generate its mean lengths for age classes.
C
C 10) Added a list of statistics pertaining to the lengths simulated
C     as well as expanding the age distribution table. The lists in
C     the table showing the percentage of fish at each age class before
C     and after recruitment, now show the effect of the recruitment
C     parameter.
C-----------------------------------------------------------------------
C
C Previous version was by R. Stanley. Comments have been retained to
C document the evolution of this program. His comments follow:
C
C Dave Fournier's original program was modified by R.Stanley
C for rockfish analysis. The changes that were incorporated are as
C follows.
C
C As of September 10/86:
C
C    (1) The parameterization of variance of length at age about
C        the mean length was made constant in subroutine GETMS.
C
C    (2) Jan, 1988, the above variance was set to a constant SDLAGE
C        read from the input file (i.e.array S(I) set to the constant
C        SDLAGE.
C
C    (3) Incorporated partial recruitment.
C      (A) In subroutine GETPS, the calculation of population
C          proportion at age is alterred deterministically
C          by multiplying the real proportion by a partial
C          recruitment factor for each age.
C
C      (B) Subroutine NASIM was re-written to read in the
C          partial recruitment parameters in one additional
C          line of the input file. Parameters are
C          written in free format starting with value
C          corresponding to age of first recruitment
C          (ie. 0.015 0.080 0.279 0.59 0.95 N*1.0, where
C          N is a repeat specifier for the data, and is
C          large enough to specify recruitment parameters
C          up to the maximum number of age classes to be
C          used)
C
C  4) There appeared to be a bug in Dave's logic such
C     that the biggest possible mean length at age could
C     not exceed the mean length of the oldest age. Given
C     variance in mean length at age this is clearly
C     not the case so modified the variable L2 to be
C     equal to the biggest mean length at age.
C
C  5) Set the std. dev. of length at age to be a constant.
C
C-----------------------------------------------------------------------
C                    Column reference scale
C       1         2         3         4         5         6         7-7
C...+.7..0....+....0....+....0....+....0....+....0....+....0....+....0.2
C-----------------------------------------------------------------------
C
C     Common block /LGUNIT/ variables.
C                 INTKEY --- logical unit for interactive keyboard input
C                 INTCRT --- logical unit for interactive display (CRT)
```

```
C               IUNITO --- logical unit for output from simulation
C
C       Common block /OVBLK/ variables.
C             Integers scalars:
C                                MAXAG  --- maximum age in simulation
C                                MINAG  --- minimum age in simulation
C                                MXFREQ --- limit to the number of length
C                                           intervals for distribution
C                                MXYRS  --- limit to the number of
C                                           years/age-classes for the
C                                           simulation
C                                NAGE   --- number of ages simulated
C                                NFREQ  --- number of length frequencies
C                                NLONG  --- number of fish longer than
C                                           lengths of interest
C                                NSHORT --- number of fish shorter than
C                                           lengths of interest
C
C             Real scalars     :
C                                L1     --- est. smallest length of fish
C                                L2     --- est. largest length of fish
C                                LINT   --- interval for length distrib.
C                                POPORG --- normalization factor for the
C                                           population of the original
C                                           fish before recruitment
C                                POPSUM --- normalization factor for the
C                                           population of recruitable
C                                           fish
C                                POPSUM --- total of recruitable fish
C                                RSIG   --- recruitment std. deviation
C                                VBK    --- von Bertalanffy k constant
C
C             Real arrays      :
C                                F      --- length frequency distributions
C                                M      --- mean length at age
C                                P      --- rel. no. in population at age
C                                PREC   --- partial recruitment parameters
C                                S      --- std. dev. of length at age
C                                SAMORT --- sum of the annual mortality
C                                           rates from MINAG up to the
C                                           age of each of the NAGE age
C                                           classes.  Includes the effects
C                                           of natural mortality, fishing
C                                           mortality and the extent of
C                                           recruitment of the species.
C                                           (Mortalities are specified or
C                                           read from a mortality schedule
C                                           file (eg. MORTALITY.LAS))
C                                FYEAR  --- fishing mortality rate for
C                                           each year from the present
C                                           time back to the time that
C                                           the oldest age class was
C                                           recruited.
C                                MAGE   --- natural mortality rate for
C                                           each age class.
C             Character scalar:
C                                TITLE  --- type of fish being examined
C                                           (80 characters, maximum)
C
C       Parameters set below and in most subroutines:
C                   MAXFRQ --- max. number of length intervals for
C                              length-frequency distribution
C                   MAXYRS --- max. number of years to simulate ages
C                              of fish
C                   IKEY   --- logical unit for interactive keyboard input
C                   ICRT   --- logical unit for interactive display (CRT)
C                   IOUT   --- logical unit for output from simulation
C
C        Name of the output file set in subroutine ASSIGN:
C
C                   OUTFLE --- name of the file to use for output from
```

```
C                               simulation.  (Filename.extension form,
CC                              maximum of 20 characters).  If using VAX
CCC                             VMS, a version number of +1 can be added,
CCCC                            to automatically create a new version, and
CCCCC
CCC-------------------------------------------------------------------------
CC
C
      CHARACTER OUTFLE*20, OUTFIL*20
      PARAMETER (MAXFRQ = 100,
     &           MAXYRS = 150,
     &           IKEY = 5,
     &           ICRT = 6,
     &           IOUT = 20)
C
      LOGICAL LASTFL
      REAL    M, MAGE
      REAL    L1, L2, LINT
      CHARACTER TITLE*80, MTYPE*20
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C
      COMMON /OVBLK/ MAXAG, MINAG, MXFREQ, MXYRS, NAGE, NFREQ, NLONG,
     &   NSHORT,
     &   L1, L2, LINT, POPORG, POPSUM, RSIG, VBK,
     &   F(MAXFRQ),
     &   M(MAXYRS), P(MAXYRS), PREC(MAXYRS), S(MAXYRS), SAMORT(MAXYRS),
     &   FYEAR(MAXYRS), MAGE(MAXYRS),
     &   TITLE
C
      DATA NLONG, NSHORT/0,0/
C
C     Set variables equal to parameters for maximum freq. and years.
      MXFREQ = MAXFRQ
      MXYRS  = MAXYRS
      INTKEY = IKEY
      INTCRT = ICRT
      IUNITO = IOUT
C
      WRITE(INTCRT,9000)
9000  FORMAT(/,' Length-Frequency Distribution Simulator. ',
     &   ' (DFO, 1988)',/,
     &   ' ------------------------------------------',
     &   '----------')
C
C     Assign the appropriate file name to OUTFLE.
      CALL ASSIGN(OUTFLE,LASTFL)
C
      WRITE(IUNITO,9000)
      WRITE(INTCRT,9010) OUTFLE
9010  FORMAT(/,' Program output for this run will be written to file: '
     &   ,A20,/)
C
C     Simulate age and length of NFISH fish in population.
      CALL NASIM(FLAG,MTYPE,NFISH)
C
C     Print message if any simulated lengths were outside of the range
C     of interest.
      CALL NASMCK(L1, L2, NLONG, NSHORT)
C
C     Print results.
      CALL LPOUT(OUTFLE)
C
C     Print the length-frequency distribution curve on a line-printer.
      CALL LPPLOT(FLAG,NFISH)
C
C     Save the distribution for this run, concatenated to the data sets
C     saved from previous runs.  Write message to OUTFLE, to describe
C     the file used.
      CALL DSSAVE(LINT,L1,L2,F,MTYPE,OUTFLE,NFREQ)
```

```
C
C     Close output file.
      CALL ASSGCL(OUTFLE,LASTFL)
C
      STOP ' '
      END
C*****************************************************************************
C
      SUBROUTINE ASSIGN(OUTFLE,LASTFL)
C
C-----------------------------------------------------------------------------
C
C     Assigns a file name to OUTFLE to be used for the printed output
C     from the simulation.  The list of available file names below is
C     scanned, and the first name that will be a new file is used.
C     Opens OUTFLE. If all files already exist, then program stops.
C
C     Sets LASTFL to true if all output files will contain data after
C     this run.
C
C     Has entry point ASSGCL to close file OUTFLE.
C-----------------------------------------------------------------------------
C
C     Set the number of files available for output:
C     (Note:- Formats 9000 and 9040 to be specified
C     to print NFILEO file names)
      PARAMETER (NFILEO = 5)
C
C
      LOGICAL FEXIST, LASTFL
      CHARACTER OUTFLE*20, FILEN(NFILEO)*20
C
C     Define the names of the possible output file:
      DATA      FILEN(1) / 'LFMFILE.OU1             '/,
     &   FILEN(2) / 'LFMFILE.OU2             '/,
     &   FILEN(3) / 'LFMFILE.OU3             '/,
     &   FILEN(4) / 'LFMFILE.OU4             '/,
     &   FILEN(5) / 'LFMFILE.OU5             '/
C
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C
C     Find the first available file in the above list.
      DO 20 INQ=1,NFILEO
         INQUIRE(FILE=FILEN(INQ), EXIST=FEXIST, ERR=20)
         IF(FEXIST) THEN
C           File previously existed.  Continue looking for an
C           available file.
         ELSE
C           Suitable file found.
            OUTFLE = FILEN(INQ)
            DO 10 I=INQ+1,NFILEO
               INQUIRE(FILE=FILEN(I), EXIST=FEXIST, ERR=10)
               IF(.NOT. FEXIST) THEN
C                 A file name is available for the next run.
                  LASTFL = .FALSE.
C                 Open a new output file for each program run:
                  OPEN(UNIT=IUNITO, FILE=OUTFLE, STATUS='NEW',ERR=30)
                  RETURN
               END IF
10          CONTINUE
C           Output file stack will be full after this run.
            LASTFL = .TRUE.
C           Open a new output file for each program run:
            OPEN(UNIT=IUNITO, FILE=OUTFLE, STATUS='NEW',ERR=30)
            RETURN
         END IF
20    CONTINUE
C
C     No suitable file was found.
```

```
      WRITE(INTCRT,9000) ((I,FILEN(I)),I=1,NFILEO)
 9000 FORMAT(/,' ************ Program Run Cancelled ************',
     &    /,T3,' Creating a new output file for this program ',
     &    /,T3,' will overwrite output from previous runs.',
     &    /,T3,' Please delete or rename some of the following ',
     &    /,T3,' files before rerunning this program.',
     &    //,T3,' Note:- The program will use the first unused',
     &    /,T3,' file name from this list.',
     &    //,' Order of Search',T25,'File Name',
     &    5(/,1X,T8,I2,T25,A20))
      WRITE(INTCRT,9010)
 9010 FORMAT(//,' Program cancelled')
      STOP
C
 30   CONTINUE
      WRITE(INTCRT,9020) OUTFLE
 9020 FORMAT(/,' *** ASSIGN: Error on call to open file: ',A20,' ***',/)
      STOP
C
CC------ Entry point to subroutine ASSIGN ------------------------------
C
      ENTRY ASSGCL(OUTFLE,LASTFL)
C
CC---------------------------------------------------------------------
CCC
CCC   Close file opened by the call to ASSIGN, and print message if
CCC   output stack will be full after this run.
CCC
CC----------------------------------------------------------------------
CC
C
      IF(LASTFL) THEN
C        Output stack will be full after this run.
         WRITE(INTCRT,9030) OUTFLE,((I,FILEN(I)),I=1,NFILEO)
         WRITE(IUNITO,9030) OUTFLE,((I,FILEN(I)),I=1,NFILEO)
 9030    FORMAT(//,62('*'),//,' WARNING:- Output from this run,'
     &        ' written to file: ',A20,
     &      /,T11,' All output file names have now been used.'/
     &      /,T11,' Please delete or rename some of the following',
     &      /,T11,' files before rerunning this program.',
     &      //,T11,' Note:- The program will use the first unused',
     &      /,T11,' file name from this list.',
     &      //,' Order of Search',T25,'File Name',
     &      5(/,1X,T8,I2,T25,A20),//,62('*'),/)
C
      ELSE
         WRITE(INTCRT,9040) OUTFLE
         WRITE(IUNITO,9040) OUTFLE
 9040    FORMAT(/,' Program output for this run has been written to ',
     &        'file: ',A20,//,' End of program LFMODEL',//,
     &      1X,79('*'))
      END IF
C
      CLOSE(UNIT=IUNITO, STATUS='KEEP')
      RETURN
      END
CC**********************************************************************
C
      SUBROUTINE DSSAVE(LINT,L1,L2,F,MTYPE,OUTFLE,NFREQ)
C
CCC   Subroutine to save the length-frequency distribution.
CCC
CC----------------------------------------------------------------------
CCC
CCC   Calling sequence:
CCC                       LINT   --- length interval for the distribution
CCC                       L1     --- smallest length category
CCC                       L2     --- largest length category
CCC                       F      --- length distribution array
CCC                       MTYPE  --- 20 character message to
C                                    indicate the mortality type used
```

```
C                       OUTFLE --- name of the file used for the
C                                  output from the simulation
C                       NFREQ  --- number of length categories in the
C                                  distribution
C
C     The output from this run will be added to the end of this
C     file if there are fewer than MAXSAV previously saved outputs.
C     If the number already saved is MAXSAV (or more), then this
C     output will start at the MAXSAV position, and will become the
C     last data set in the file.  A warning will be printed to units
C     INTCRT and IUNITO, to indicate that old data would will be lost
C     with this run or with the next run of this program.
C
C     If an error occurs while reading the output from previous runs,
C     then the subroutine will write to an alternate file, which will
C     contain output for only this run.
C
C     Parameters:
C                 MAXSAV --- max. number of data sets to save per file.
C                 IUNTP  --- unit number of file for concatenated data.
C                 IUNITD --- unit number of file to use to "dump" a
C                            single data set if unable to use IUNTP.
C                 PRFILE --- name of file for concatenated data sets.
C                 DMPFLE --- name of file to use if "dumping" a data set.
C-----------------------------------------------------------------------
C
      CHARACTER LINE*80, PRFILE*20, PFILE*20, DMPFLE*20, MTYPE*20,
     &   OUTFLE*20
      REAL L, LINT, L1, L2, F(*)
      PARAMETER (MAXSAV = 5,
     &           MXSAV1 = MAXSAV - 1,
     &           IUNTP  = 21,
     &           IUNITD = 22,
     &           PRFILE = 'LFMFILE.PRO          ',
     &           DMPFLE = 'LFMFILE.DMP          ')
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
CC
CC    Copy file name and unit number to use for output.
      PFILE = PRFILE
      IUNITP = IUNTP
C
CC    Set default run number if file is empty.
      NUMRUN = 0
      NREAD = 0
C
CCC   Open file to retain a summary of the distributions for
CC    processing by subsequent programs.
      OPEN(UNIT=IUNITP, FILE=PFILE, STATUS='UNKNOWN', ERR=80)
      WRITE(IUNITO,9000) PFILE
 9000 FORMAT(
     &   '1',/,' DATA SET HEADINGS IN DISTRIBUTION DATA SUMMARY FILE: ',
     &   A20,/,
     &   ' ---------------------------------------------------------',
     &   11('-'),//)
      DO 50 NRUNS=1,MXSAV1
C        Read output file header section (12 lines)
C        Note: The first line of the header contains '1' as the
CCCCC          first character.  The other 11 lines of the header begin
CCCCC          with a blank.  The variable LINE will read from column 2
CCCCC          up to column 81 from the input file, so, as a result,
C              the column tabs for reading the information from LINE
C              are one less than those for writing to the file.
         DO 10 NSKIP=1,3
C           Read and print the first 3 line of the header from
CC          each previous run.
            READ(IUNITP,9010,END=60,ERR=80) LINE
            WRITE(IUNITO,9010) LINE
 9010       FORMAT(1X,A80)
 10      CONTINUE
```

```
C        Read the value of NFREQ and LINT used for the past run.
         READ(LINE,9020,ERR=80) NFREQP, PLINT
9020     FORMAT(T32,I2,T38,F4.1)
C        Read and print the next 4 lines from the header (lines 4 to 7).
         DO 20 NSKIP=1,4
            READ(IUNITP,9010,END=80,ERR=80) LINE
            WRITE(IUNITO,9010) LINE
20       CONTINUE
C        Read the run number from the last LINE read.
         READ(LINE,9030,ERR=80) NUMRUN
9030     FORMAT(T58,I2)
         DO 30 NSKIP=1,4
C           Header lines 8, 9, 10 and 11.
            READ(IUNITP,9010,END=80,ERR=80) LINE
            WRITE(IUNITO,9010) LINE
30       CONTINUE
C
C        Skip the data plus the blank line at the end of the header.
         DO 40 I=0,NFREQP
            READ(IUNITP,9010,END=80,ERR=80) LINE
40       CONTINUE
         WRITE(IUNITO,9040)
9040     FORMAT(//)
50    CONTINUE
C
C     File contains at least MXSAV1 sets of data.  When the data
C     from this run is written, the maximum will have been reached.
      WRITE(IUNITO,9050) PFILE, MAXSAV, NUMRUN + 1
      WRITE(INTCRT,9050) PFILE, MAXSAV, NUMRUN + 1
9050  FORMAT(//,62('*'),//,
     &    ' WARNING:- The maximum number of data sets on file ',
     &    A20,/,' has been reached.  Old data sets will be lost',
     &    ' now or with the',/,' next run of this program.',/,
     &    ' Please remove any unwanted data sets and/or rename',
     &    ' the file.',/,' Next run will overwrite the ',I2,
     &    '-th data only (i.e. Run no. 'I2,')',//,62('*'),/)
C
60    CONTINUE
C     Write the summary of the distributions for this run.
C     Write data set headers to IUNITO as well.
      IF(IUNITP .NE. IUNITD) THEN
         WRITE(IUNITO,9060) PFILE
         WRITE(INTCRT,9060) PFILE
9060     FORMAT(/,' Length-frequency distribution is being added',
     &          ' to file: ',A20)
      ELSE
         WRITE(IUNITO,9070) PFILE
         WRITE(INTCRT,9070) PFILE
9070     FORMAT(/,' Length-frequency distribution is being written',
     &          ' to alternate file: ',A20)
      END IF
      WRITE(IUNITO,9080)
9080  FORMAT(' with the data set heading as below:',/)
C
      NUMRUN = NUMRUN + 1
C
      WRITE(IUNITP,9090) MTYPE
9090  FORMAT('1' 'LFMODEL.FOR summary for ..... ',T33,A20)
      WRITE(IUNITO,9100) MTYPE
9100  FORMAT(' ' 'LFMODEL.FOR summary for ..... ',T33,A20)
      WRITE(IUNITP,9110) OUTFLE, NFREQ, LINT, NUMRUN, LINT/2.0
      WRITE(IUNITO,9110) OUTFLE, NFREQ, LINT, NUMRUN, LINT/2.0
9110  FORMAT(    ' For details see file ..... ',T33,A20,/,
     &    T2, 'No. of values after header ... ',T33,I2,
     &    ' at ',T39,F4.1,' cm intervals',/,
     &    T2, 'To read above values ..... ',T33,
     &    'format(T33,I2,T39,F4.1)',/,
     &    T2, 'To read data use ..... ',T33,
     &    'format(T2,I2,T5,F10.1,T20,I12)',//,
     &    T2, 'Distribution of Fish in Sample Population',
     &    ' for Run Number ',T59,I2,/,
     &    T2, '-----------------------------------------',
```

```
     &   '------------------',//,
     &   ' Run',T10,'Length',T25,'Frequency',/,
     &   ' No.',T8,'+/-',T12,F3.1,' cm',//)
      DO 70 I=1,NFREQ
         L = L1+(REAL(I)-1.0)*LINT
         WRITE(IUNITP,9120) NUMRUN,L, INT(F(I))
9120     FORMAT(' ',I2,T5,F10.1,T20,I12)
70    CONTINUE
C
      CLOSE(UNIT=IUNITP, STATUS='KEEP')
C     Hold screen until users responds with a carriage return.
      CALL HOLD(INTCRT,INTKEY)
      RETURN
C
80    CONTINUE
C     Error from reading PFILE.  Write to file DMPFLE instead.
      CLOSE(UNIT=IUNITP  STATUS='KEEP')
      WRITE(IUNITO,9130) PFILE, DMPFLE
      WRITE(INTCRT,9130) PFILE, DMPFLE
9130  FORMAT(//,' WARNING:- Abnormal read or unexpected end on file ',
     &   A20,
     &   /,' File now closed.  Using alternate summary file. ',
     &   A20,//)
      PFILE = DMPFLE
      WRITE(IUNITO,9000) PFILE
      IUNITP = IUNITD
      OPEN(UNIT=IUNITP, FILE=PFILE, STATUS='UNKNOWN', ERR=90)
      GO TO 60
C
C
90    CONTINUE
      WRITE(IUNITO,9140) PFILE
      WRITE(INTCRT,9140) PFILE
9140  FORMAT(/,' ERROR:- Error when opening file: ',A20,/,
     &   ' No length-distribution summary file written.',/)
C
      CLOSE(UNIT=IUNITP, STATUS='KEEP')
C     Hold screen until users responds with a carriage return.
      CALL HOLD(INTCRT,INTKEY)
      RETURN
      END
C
C*******************************************************************************
C
      SUBROUTINE FGEN (P, NAGE, IAGE, M, S, IL, NFREQ ,IXO, L1, L2 , L,
     &   LINT, NLONG, NSHORT)
C
C-------------------------------------------------------------------------------
C
C  This subroutine generates a fish of a certain age,IAGE and length,L
C  and assigns it to a length category, IL.  If the length is either too
C  long or too short for the range of the length categories, then IL
C  is set to minus one (-1).  P is the relative abundance of fish in
C  each of the NAGE age classes, of mean length,M, with std. dev. of
C  length about M(i) of S(i).  IXO is a random seed.  Distribution
C  will have NFREQ length categories from L1 to L2, intervals of LINT.
C  NSHORT and NLONG are the numbers of fish outside of the range of
C  distrbution.
C
C-------------------------------------------------------------------------------
C
      REAL*8 IXO
      REAL L1, L2, L, LINT
      REAL M(*), S(*), P(*)
C
C  First assign the fish to an age class.
C     Use a pseudo-random number in the range of 0 to 1.0, to choose
C     the relative position in the population.
      X = RANNUM(IXO)
C
      SUM = 0.
      DO 10 J= 1,NAGE
```

```
            SUM = SUM + P(J)
            IF(X .LE. SUM) THEN
C              Youngest age class found where at least X of the population
C              is this age or younger.
               IAGE = J
               GO TO 20
            END IF
10       CONTINUE
C
C     Then generate a normal random fish length suitable for this age
C     class.
20       CONTINUE
C        Determine two pseudo-random numbers in the range of 0 to 1.0.
         X = RANNUM(IXO)
         Y = RANNUM(IXO)
C
C        Determine a random number from a normal distribution (zero mean).
         RN = SQRT(-2.*ALOG(X))*COS(2.*3.1415926*Y)
C
C        Determine a random length for fish in this age class from
C        the mean length (M) and the standard deviation (S) in the
C        mean length for this age.
         L    = M(IAGE) + RN*S(IAGE)
C
C     Finally put the observed length into correct frequency slot, keeping
C     track of the number of fish outside of the range of interest.
C     Note:- IL-th length category ipcludes lengths  greater than or
C            or equal to [L1 + (IL-1)*LINT] - 0.5*LINT but
C            less than   [L1 + (IL-1)*LINT] + 0.5*LINT.
C
         DELLEN = 0.5*LINT
         IF(L .LT. (L1-DELLEN)) THEN
C           Fish is shorter than lengths of interest.
            NSHORT = NSHORT + 1
            IL = -1
         ELSE IF( L .GE. (L2+DELLEN)) THEN
C           Fish is longer than lengths of interest.
            NLONG = NLONG + 1
            IL = -1
         ELSE
C           Set length category for fish.
            IL = INT((L-L1)/LINT + 1.5)
         END IF
         RETURN
         END
C
C*******************************************************************************
C
      SUBROUTINE GETMS(IXO,M,MINAG,NAGE,S,SDLAGE,VVB,VONBRT,VBK,VBL,VBT,
     &     A,B,ITAU1,ITAU2,Y1,Y2)
C
C-------------------------------------------------------------------------------
C
C     This subroutine sets up the mean lengths and standard deviations
C     for each age.
C
C     If VONBRT is true, it uses von Bertalanffy's growth function (also
C     called Putter No. 1).
C
C     If VONBRT is false, it uses Schnute's generalized growth function.
C
C     See the reading section of NASIM for the definition of the calling
C     sequence terms (also subr. SCHNTE).
C-------------------------------------------------------------------------------
C
      LOGICAL VONBRT
      REAL M(*), S(*)
      REAL*8 IXO
C
C
      IF(VONBRT) THEN
C        Using von Bertalanffy's growth function:
```

```
         MINAG1 = MINAG - 1
         DO 10 I=1,NAGE
C           Set mean length for age
            M(I) = VBL * (1. - EXP( -VBK*((I+MINAG1)-VBT) ) )
10       CONTINUE
      ELSE
C        Use Schnute's growth function.
         CALL SCHNTE(A,B,NAGE,MINAG,ITAU1,ITAU2,Y1,Y2,M)
      END IF
C
      DO 20 I=1,NAGE
C        Set std. dev. of length at age to be a constant for all ages.
         S(I) = SDLAGE
20    CONTINUE
C     Alter the mean lengths at age by a random (0 to 1) fraction of
C     VVB, the standard deviation in the mean length for age classes.
      IF(VVB .NE. 0.0) THEN
         DO 30 I=1,NAGE
C           Determine two pseudo-random numbers in the range of 0 to 1.
            X = RANNUM(IXO)
            Y = RANNUM(IXO)
C
C           Determine a random number from a normal distribution
C           (zero mean).
            RN = SQRT(-2.*ALOG(X))*COS(2.*3.1415926*Y)
C
C           Add the std. dev. in mean length at age I + MINAG - 1,
C           where a normal distribution is applied to VVB.
            M(I) = M(I) + RN * VVB
30       CONTINUE
      END IF
      RETURN
      END
C
C*******************************************************************************
C
      SUBROUTINE GETPS(NAGE,IXO,SAMORT,RSIG,P,PREC,POPORG,POPSUM)
C
C-------------------------------------------------------------------------------
C
C  This subroutine calculates the relative number of fish in each age
C  class using SAMORT(I), the accumulated mortality rate that the
C  I-th age class has experienced since its recruitment age.  The
C  accumulated rate includes the individual effects of natural morality,
C  fishing mortality, and partial recruitment, experienced by the I-th
C  age class of fish from its recruitment age to the present time.
C
C  Define SAMORT(I) =  [Sum of PREC(K)*FYEAR(I-K+1)] + [Sum of MAGE(K)]
C                      for K = 1 to I.
C
C            The sum of MAGE(K) for  K=1 to I, is the natural mortality
C            rate experienced by fish of the I-th age class, from their
C            recruitment age up to their present age.
C
C            FYEAR(L) is the fishing mortality rate for (L-1) years ago,
C            where L = 1,2,3,....,I and FYEAR(1) is the fishing mortality
C            rate this year, and FYEAR(I) is the rate when the I-th age
C            class was recruited.
C
C            PREC(K) is the partial recruitment parameter for fish of
C            the K-th age class.
C
C  For example fish in the 5-th age class have survived the following:
C     i) Natural mortality =
C           MAGE(5) for their present age, this year
C         + MAGE(4) for their age, last year
C         + MAGE(3) for their age, the year before that
C         + ....
C         + MAGE(1) for their age, when recruited
C    ii) Fishing mortality =
C           FYEAR(1)*PREC(5) for this year when they are in age class 5
C         + FYEAR(2)*PREC(4) for last year when they were in age class 4
```

```
C          + FYEAR(3)*PREC(3) for year before, when they were age class 3
C          + .....
C          + FYEAR(5)*PREC(1) for the year when they were first recruited
C-----------------------------------------------------------------------
C
      REAL*8 DXD
      REAL   P(*), PREC(*), SAMORT(*)
C
      POPORG = 0.
      POPSUM = 0.
      DO 10 I=1,NAGE
C        Determine two pseudo-random numbers in the range of 0 to 1.0.
         X = RANNUM(DXD)
         Y = RANNUM(DXD)
C
C        Determine a random number from a normal distribution
C        (zero mean).
         RN = SQRT(-2.*ALOG(X))*COS(2.*3.1415926*Y)
C
C        Population of the I-th age class considering partial recr'tm'pt
C        & its standard deviation and the accumulated mortality (SAMORT)
C        for the last I-years.
         PORIG = EXP(RSIG*RN - SAMORT(I))
         P(I)  = PORIG * PREC(I)
C
C        Total population considering partial recruitment.
         POPSUM = POPSUM + P(I)
C
C        Determine the sum in the original population.
         POPORG = POPORG + PORIG
10    CONTINUE
C
      DO 20 I=1,NAGE
C        Relative population in the I-th age class.
         P(I) = P(I) / POPSUM
20    CONTINUE
C
      RETURN
      END
C
C*************************************************************************
C
      SUBROUTINE HOLD(INTCRT,INTKEY)
C
C     Hold screen until user responds with a carriage return.
C     Message written to unit INTCRT. Response is read from unit INTKEY.
C-----------------------------------------------------------------------
C
      CHARACTER RET*1
C
      WRITE(INTCRT,9000)
9000  FORMAT(/,' Press RETURN to continue ...')
      READ(INTKEY,9010) RET
9010  FORMAT(A1)
      RETURN
      END
C
C*************************************************************************
C
      SUBROUTINE LPOUT(OUTFLE)
C
C     Subroutine to print results to unit IUNITO, with name OUTFLE.
C-----------------------------------------------------------------------
C     Parameters set below and in most subroutines:
C               MAXFRQ  --- max. number of length intervals for
C                           length-frequency distribution
C               MAXYRS  --- max. number of years to simulate ages
C                           of fish
```

```
C
C       PARAMETER (MAXFRQ = 100, MAXYRS = 150)
C
C
        REAL    M, MAGE
        REAL    L1, L2, LINT, L
        CHARACTER TITLE*80, OUTFLE*20
C
C       COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
        COMMON /OVBLK/ MAXAG, MINAG, MXFREQ, MXYRS, NAGE, NFREQ, NLONG,
     &    NSHORT,
     &    L1, L2, LINT, POPORG, POPSUM, RSIG, VBK,
     &    F(MAXFRQ),
     &    M(MAXYRS), P(MAXYRS), PREC(MAXYRS), S(MAXYRS), SAMORT(MAXYRS),
     &    FYEAR(MAXYRS), MAGE(MAXYRS),
     &    TITLE
C
C
C       Check if parameters for maximum freq. and years agree with main.
        IF(MAXFRQ .NE. MXFREQ .OR. MAXYRS .NE. MXYRS) THEN
           CALL PARERR(MAXFRQ, MXFREQ, MAXYRS, MXYRS)
           STOP 'See subroutine LPOUT'
        END IF
C
        WRITE(IUNITO,9000)
9000    FORMAT('1' //,T17,
     &      ' AGE DISTRIBUTION OF FISH IN SAMPLE POPULATION',
     &    T83,'MORTALITY RATE COMPONENTS',/,
     &    T17,'-------------------------------',
     &    T83,'-----------------------------------------------',
     &    /,'   Age',T13,'Mean',T20,'Act''l Abund.',T34,'% at Age of',
     &    T47,'Part Recr.',T58,'Z, Accum''ltd',T72,'EXP(-Z)',
     &    T83,'Natural',T97,'Fishing',
     &    /,T12,'Length',T20,'Before Recr.',T33,'Fishable Pop.',
     &    T49,'Factor',T59,'Mort. Rate',
     &    T84,'Rate',   T92,'Years ago  Rate',/)
C
        PFACTR = POPORG/POPSUM
        DO 10 I=1,NAGE
           II = I + MINAG - 1
C          Percentage at age of fishable population.
           P(I) = 100.* P(I)
C          percentage at age of population before recruitment.
           XXX = P(I)/PREC(I)/PFACTR
C          Mortality factor without the randomness used in subr. GETPS
           YYY = EXP(-SAMORT(I))
C
           WRITE(IUNITO,9010) II,M(I),XXX,P(I),PREC(I),SAMORT(I),YYY,
     &       MAGE(I),(I-1),FYEAR(I)
10      CONTINUE
9010    FORMAT(T3,I3,T10,F8.2,T21,F8.2,T34,F8.2,T46,F8.2,T58,F8.2,T70,
     &    F8.3,T83,F5.2,T96,I3,T103,F5.2)
C
        RETURN
        END
C
C*******************************************************************************
C
        SUBROUTINE LPPLOT(FLAG,NFISH)
C
CCCCC   Subroutine to plot length-frequency distributions as a printed
CCCCC   page to be viewed on a terminal or lineprinter at least
CCCCC   (IWIDTH + 30) characters wide.  The absicissa is vertically
CCCCC   down the page, while the ordinate is horizontal from left to
CCCCC   left on the page.  The numerical value of each ordinant is
CCCCC   printed at the right of the plot, while the scaled position
CCCCC   on the plot is indicated by an asterisk (*).  The zero position
CCCCC   of the absicissa is indicated by a vertical line (|) unless
CCCCC   an ordinate value falls at this position.
C       Draws on the absicissa at the value of "FLAG" to act as a
```

```
C     reference.  Line is drawn only if "FLAG" is within the range of
CC    the data.
CCC
CCC   Output is to unit INTCRT (i.e. terminal if an interactive run)
CCC   and to unit IUNITO.
CCC---------------------------------------------------------------------
CCC
CCC   PARAMETERS:
CCC
CCC   Define: IWIDTH -- the 'plot' width,
CCC           FMAX   -- the upper limit of the ordinate,
CCC           FPOS   -- .true. if function to plot is all positive.
CCC                     .false., then will use only half of IWIDTH
CCC                     for the positive portion of the plot.
CCC
CCC   Note:- the length of the line in the format 9030 (i.e. nnA1) must
CC          agree with IWIDTH below.
C
      LOGICAL FPOS
      PARAMETER (IWIDTH = 50
     &           FMAX   = 1.0E35,
     &           FPOS   = .TRUE.)
C
CCC   Parameters set below and in most subroutines:
CCC             MAXFRQ  --- max. number of length intervals for
CCC                         length-frequency distribution
CCC             MAXYRS  --- max. number of years to simulate ages
CCC                         of fish
C
      PARAMETER (MAXFRQ = 100, MAXYRS = 150)
C
CCC---------------------------------------------------------------------
C
      REAL    M, MAGE
      REAL    L1, L2, LINT, L
      CHARACTER TITLE*80, LCHAR(IWIDTH)*1
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
      COMMON /OVBLK/ MAXAG, MINAG, MXFREQ, MXYRS, NAGE, NFREQ, NLONG,
     &   NSHORT,
     &   L1, L2, LINT, POPORG, POPSUM, RSIG, VBK,
     &   F(MAXFRQ),
     &   M(MAXYRS), P(MAXYRS), PREC(MAXYRS), S(MAXYRS), SAMORT(MAXYRS),
     &   FYEAR(MAXYRS), MAGE(MAXYRS),
     &   TITLE
C
CCC   Check if parameters for maximum freq. and years agree with main.
      IF(MAXFRQ .NE. MXFREQ .OR. MAXYRS .NE. MXYRS) THEN
         CALL PARERR(MAXFRQ, MXFREQ, MAXYRS, MXYRS)
         STOP 'See subroutine LPPLOT'
      END IF
C
CCC   Set scaling parameters, using the appropriate scaling for
CCC   the function to be plotted.
C
      IF(FPOS) THEN
C        For a function with all positive values:
         ISCALE = IWIDTH - 1
         IAXIS  = 1
      ELSE
C        For a function with both positive and negative values:
C        (with X-axis in the middle of the page)
         ISCALE = (IWIDTH - 1)/2
         IAXIS  = ISCALE + 1
      END IF
C
      ABSP   = 0.0
```

```
      DO 10 I=1,NFREQ
C        Find the maximum range in the function to plot.
         FF = ABS( F(I) )
         IF(FF .LT. FMAX .AND. FF .GT. ABSF) ABSF = FF
10    CONTINUE
      SCALE = ABSF / REAL(ISCALE)
      IF(ABSF .EQ. 0.0) THEN
C        Function f(i) contains all zeroes.  Skip plotting.
         WRITE(INTCRT,*) 'Only zeroes to plot'
         WRITE(IUNITO,*) 'Only zeroes to plot'
      ELSE
C        Plot values of F(I) where I=1 to NAGE:
         DO 20 J=1,IWIDTH
C           Blank-out line characters.
            LCHAR(J) = ' '
20       CONTINUE
C        Write title.
         WRITE(INTCRT,9000) LINT/2.0
         WRITE(IUNITO,9000) LINT/2.0
9000     FORMAT('1',//,T35,' DISTRIBUTION OF FISH IN SAMPLE POPULATION',
     &      /,T35,'----------------------------------------------',
     &      //,' Freq''ncy',T12,'Prop of',T21,'Length (cm)',
     &      T51,'Frequency'/
     &      T12,'Distrib',T22,'+/- ',F3.1,/)
C        Determine array position for flag.
         IFLAG = (FLAG - L1)/LINT  + 1.0
C        Calculate the number of fish in the distribtution.
         NUMFSH = NFISH - NSHORT - NLONG
         DO 50 I=1,NFREQ
C           Set axis character.
            LCHAR(IAXIS) = '|'
            IF(I .EQ. IFLAG) THEN
C              At position specified by FLAG.  Mark with '-'.
               DO 30 J=2,IWIDTH
                  LCHAR(J) = '-'
30             CONTINUE
            END IF
C           Define length.
            L = L1+(REAL(I)-1.0)*LINT
            LL = INT(L)
C           Set asterisk at character position in line.
            J = INT( F(I)/SCALE + 0.5) + IAXIS
            IF(J .GE. 1 .AND. J .LE. IWIDTH) THEN
               LCHAR(J) = '*'
            END IF
            PROP = F(I) / REAL(NUMFSH)
            WRITE(INTCRT,9010) INT(F(I)), PROP, LL, LCHAR
            WRITE(IUNITO,9010) INT(F(I)), PROP, LL, LCHAR
9010        FORMAT(' ', T1,I6,T12,F4.3,T21,I5,T30,50A1)
C           Blank-out last character 'plotted'.
            LCHAR(J) = ' '
            IF(I .EQ. IFLAG) THEN
C              At position specified by FLAG.  Remove marking.
               DO 40 J=2,IWIDTH
                  LCHAR(J) = ' '
40             CONTINUE
            END IF
C
50       CONTINUE
      END IF
C
      RETURN
      END
C*******************************************************************************
C
      SUBROUTINE NASIM(FLAG,MTYPE,NFISH)
C
C     Subroutine to control the length-frequency simulation.
C
C-------------------------------------------------------------------------------
C
```

```
C     Reads information from file INFILE on unit INUNIT to define the
C     length simulation.  Uses either Schnute's generalized growth
C     formulation or von Bertalanffy's.  The user is asked which model
C     to use.  The coefficients for both are to be supplied from INFILE.
C     The default model will be von Bertalanffy's.
C
C     Calling sequence:
C                       FLAG  ---  value to flag in LPPLOT
C                       MTYPE ---  20 character message to
C                                  indicate the mortality type used
C                       NFISH ---  number of fish in the population
C
C     Calls subroutines:
C                       MORATE --- inputs mortality rate schedule
C                                  interactively.
C                       GETPS  ---
C                       GETMS  ---
C                       FGEN   ---
C
C     Subroutines GETPS, GETMS and FEGEN use random numbers.  The
C     numbers are initialized with the following integer-valued
C     seeds.  The variable, IX0 is used to retain the value of the
C     seed as updated by the random number function.  When using
C     RAN or  GGUBFS the value of IX0 is updated after each call.
C     When using RAND, IX0 is set to zero after each call, to indicate
C     that the next value in the series is to be generated.  RAND
C     internally retains the last value generated.  To allow for
C     the possible use of any of the routines (see function RANNUM),
C     IX0 should be declared as REAL*8 within the subroutine NASIM.
C
C           Integers scalars:
C                             ISEEDP --- initial random no. seed for
C                                        population of age classes.
C                             ISEEDL --- initial random no. seed for
C                                        mean lengths for age classes.
C                             ISEEDI --- initial random no. seed for
C                                        ages and lengths of individual
C                                        fish.
C           Real*8 scalar   :
C                             IX0    --- retains successive values of
C                                        the seed used by the pseudo-
C                                        random number function.
C
C
C     Parameters set below and in most subroutines:
C                 MAXFRQ  --- max. number of length intervals for
C                             length-frequency distribution
C                 MAXYRS  --- max. number of years to simulate ages
C                             of fish
C     also:
C                 INFILE  --- 20 character name to define the file
C                             with the input data
C                 INUNIT  --- unit number for the input file
C
      CHARACTER INFILE*20
      PARAMETER (MAXFRQ = 100,
     &           MAXYRS = 150,
     &           INFILE = 'LFMODEL.DAT          ',
     &           INUNIT = 10                    )
CC
      LOGICAL   VONBRT, CERROR
      REAL*8    IX0
      REAL      L1, L2, LINT, L, LMEAN, LMEDAN
      REAL      M, MAGE
      INTEGER*4 ISEEDP, ISEEDL, ISEEDI
      CHARACTER ANSWER*1, MTYPE*20, TITLE*80
CC
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
```

```
      COMMON /OVBLK/ MAXAG, MINAG, MXFREQ, MXYRS, NAGE, NFREQ, NLONG,
     &    NSHORT,
     &    L1, L2, LINT, POPORG, POPSUM, RSIG, VBK,
     &    F(MAXFRQ),
     &    M(MAXYRS), P(MAXYRS), PREC(MAXYRS), S(MAXYRS), SAMORT(MAXYRS),
     &    FYEAR(MAXYRS), NAGE(MAXYRS),
     &    TITLE
C
C
C     Check if parameters for maximum freq. and years agree with main.
      IF(MAXFRQ .NE. MXFREQ .OR. MAXYRS .NE. MXYRS) THEN
         CALL PARERR(MAXFRQ, MXFREQ, MAXYRS, MXYRS)
         STOP 'See subroutine NASIM'
      END IF
C
C
      OPEN(UNIT=INUNIT, FILE=INFILE, STATUS='UNKNOWN')
C
C  READ TITLE
      READ(INUNIT,9000) TITLE
      WRITE(IUNITO,9010) TITLE
9000  FORMAT(A80)
9010  FORMAT(' Species simulated: ',/,
     &    ' ---------------- ',/,
     &    1X,A80,//,
     &    ' Specifications: ',/,
     &    ' -------------- ',/)
C   Read number of observations to be simulated.
      READ(INUNIT,*,END=40) NFISH
C   Read # of age-classes, first age, length interval for freq'ncies,
C   and the length value to flag on the length distribution plot.
      READ(INUNIT,*)  NAGE, MINAG, LINT, FLAG
      IF(NAGE .GT. MAXYRS) THEN
         WRITE(INTCRT,9020) NAGE, MAXYRS
         WRITE(IUNITO,9020) NAGE, MAXYRS
9020     FORMAT(//,' *** ERROR in NASIM: Number of age classes ',I4,
     &       ' exceeds program limit of ',I4,'.',/)
         STOP 'NASIM: Dimensions exceeded.'
      END IF
C   Set maximum age to simulate.
      MAXAG = MINAG + NAGE - 1
C   Read von Bertalanffy's linfinity, k, tzero.
      READ(INUNIT,*) VBL, VBK, VBT
C   Read Schnute Generalized Model Coefficients.
C   NOTE:- EXPA here is EXP(-a), for a in Schnute's paper.
      READ(INUNIT,*) Y1, Y2, EXPA, B, ITAU1, ITAU2
C   Read the standard deviation for mean length at age.
      READ(INUNIT,*) VVB
C   Read the standard deviation in calculating individual lengths.
      READ(INUNIT,*) SDLAGE
C   Read the s.d. range and the initial seeds for random numbers.
      READ(INUNIT,*)  RANGE, ISEEDP, ISEEDL
C     Ensure that seeds are odd-valued integers.
      ISEEDP = 2*(ISEEDP/2) + 1
      ISEEDL = 2*(ISEEDL/2) + 1
C      Calculate a third random seed from the mean of the above seeds
C      and ensure that it is also odd-valued.
      ISEEDI = 2*INT(ISEEDP/4.0 + ISEEDL/4.0) + 1
C   Read the constant (default) annual total mortality rate ( see
C   subroutine MORATE ) and the standard deviation in the recruitment.
      READ(INUNIT,*) ZFMORT, ZNMORT, RSIG
      WRITE(IUNITO,9030) ZFMORT, ZNMORT
9030  FORMAT(' Default fishing mortality rate ',T37,F7.2,/,
     &    ' Default natural mortality rate ',T37,F7.2)
C
C Read in partial recruitment parameters.
C
      READ(INUNIT,*) (PREC(I),I=1,10)
      READ(INUNIT,*) (PREC(I),I=11,NAGE)
C
C  Determine which growth formulation to use.
      WRITE(INTCRT,9040)
```

```
9040  FORMAT(/,' Do you wish to use Schnute''s Growth Formulation? ',
     &    ' [y/n] (default=n)',/,
     &    ' (If default or n, then will use von Bertalanffy''s)')
      READ(INTKEY,9050) ANSWER
9050  FORMAT(A1)
      IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') THEN
C        Use Schnute's model.
         VONBRT = .FALSE.
C        Schnute's growth model will be used if coefficients are valid.
         IF(Y2 .GT. Y1 .AND. Y1 .GT. 0.0 .AND. ITAU2 .GT. ITAU1) THEN
            CERROR = ' '
            WRITE(INTCRT,9060)
            WRITE(IUNITO,9060)
9060        FORMAT(/,' --- USING SCHNUTE''S FORMULATION ---')
         ELSE
C           Coefficients are not valid.  Set error message flag.
            CERROR = 'S'
         END IF
      ELSE
C        Default is to use von Bertalanffy's model.
         VONBRT = .TRUE.
C        von Bertalanffy's growth model will be used if coefficients
C        are valid.
         IF(VBL .NE. 0.0 .AND. VBK .NE. 0.0) THEN
            CERROR = ' '
            WRITE(INTCRT,9070)
            WRITE(IUNITO,9070)
9070        FORMAT(/,' --- USING VON BERTALANFFY''S FORMULA ---')
         ELSE
C           Coefficients are not valid.  Set error message flag.
            CERROR = 'V'
         END IF
      END IF
C
C  Read the schedule of annual mortality rates and
C  calculate their sums for each age from MINAG to MAXAG.
C  Write schedule to IUNITO to document the simulation.
C
      IF(CERROR .EQ. ' ') THEN
C        No error on coefficients for growth simulation.
         CALL MORATE(FYEAR,MAGE,MINAG,MAXAG,MTYPE,NAGE,PREC,SAMORT,
     &      ZFMORT,ZNMORT)
      END IF
C
C Write the input parameters.
C
      WRITE(IUNITO,9080)
9080  FORMAT(/,' Constants for Growth Models:',/)
      WRITE(IUNITO,9090) VBL, VBT, VBK
9090  FORMAT(' a) von Bertalanffy Model (Putter No. 1) :',/
     &   T5,' L infinity (cm)',T28,F16.4,/,
     &   T5,' T zero ',T28,F16.4,/,
     &   T5,' Growth constant,K',T28,F16.4,/)
C
C     Use MINAG as the starting age for Schnute model and calculate
C     the equivalent "a" for Schnute's original formulation.
      SA = -LOG(EXPA)
      WRITE(IUNITO,9100) Y1, Y2, EXPA, SA, B, ITAU1, ITAU2
9100  FORMAT(' b) Schnute Generalized Model: ',/,
     &   T5,' Y1, length (cm) at age, TAU1 ',T34,F10.4,/,
     &   T5,' Y2, length (cm) at age, TAU2 ',T34,F10.4,/,
     &   T5,' EXPA, 1st growth coefficient ',T34,F10.4,/,
     &   T5,'   (Schnute''s a=-log(EXPA)=   ',T34,F10.4,')',/,
     &   T5,' B, 2nd growth coefficient    ',T34,F10.4,/,
     &   T5,' TAU1, age (yrs) at length Y1 ',T34,I10,/,
     &   T5,' TAU2, age (yrs) at length Y2 ',T34,I10,/)
C
      WRITE(IUNITO,9110) NFISH
9110  FORMAT(' Sample size',T28,I16)
      WRITE(IUNITO,9120) NAGE
9120  FORMAT(' Number of age classes',T28,I16)
      WRITE(IUNITO,9130) MINAG, RSIG
```

```
9130  FORMAT(' Minimum age in simulation',T28,I16,
     &     /,' Standard deviation for recruitment ',T37,F7.2,/)
      WRITE(IUNITO,9140) VVB
9140  FORMAT(' Std. dev. in mean length for ages ',T37,F7.2)
      WRITE(IUNITO,9150) SDLAGE
9150  FORMAT(' Std. dev. in individual lengths ',T37,F7.2)
      WRITE(IUNITO,9160) RANGE
9160  FORMAT(' Range factor for length std. dev.',T37,F7.2)
      WRITE(IUNITO,9170) ISEEDP, ISEEDL, ISEEDI
9170  FORMAT(' Initial seed, population @ age  ',T34,I10,/,
     &     ' Initial seed, mean length @ age ',T34,I10,/,
     &     ' Initial seed, indiv. age-length ',T34,I10)
C
C  Stop simulation if an error was previously detected in coeff.
      IF(CERROR .EQ. 'S') THEN
         STOP ' NASIM: Schnute coefficients are not valid.'
      ELSE IF(CERROR .EQ. 'V') THEN
         STOP ' NASIM: von Bertalanffy coefficients are not valid.'
      END IF
C
C  Calculate the age distribution for the population.
C  Use ISEEDP as the initial seed for the age distribution.
C
      IX0 = ISEEDP
      CALL GETPS(NAGE,IX0,SAMORT,RSIG,P,PREC,POPORG,POPSUM)
C
C  Calculate the mean length for each age class.
C  Use ISEEDL as the initial seed for the length distribution.
C
      IX0 = ISEEDL
      CALL GETMS(IX0,M,MINAG,NAGE,S,SDLAGE,VVB,VONBRT,VBK,VBL,VBT,
     &   EXPA,B,ITAU1,ITAU2,Y1,Y2)
C
C This model sets the means according to von Bertalanffy or Schnute
C growth formulations.
C
C The smallest and largest lengths expected are:
C     Initial values:
      L2 = M(NAGE) + RANGE*S(NAGE)
      L1 = M(1) - RANGE*S(1)
C     Final values to nearest whole value:
      IL1 = INT(L1)
      L1  = REAL(IL1)
      T2  = AINT(L2)
      IF(L2 .NE. T2) THEN
C        Upper length limit is not an integer.  Set to next integer.
         IL2 = INT(L2) + 1
      ELSE
         IL2 = INT(L2)
      END IF
      L2  = REAL(IL2)
C
C     Determine the number of frequencies to span the L1 to L2 range.
      NFREQ = (IL2 - IL1) / LINT + 1.0
      IF(NFREQ .GT. MAXFRQ) THEN
         WRITE(INTCRT,9180) NFREQ, MAXFRQ
         WRITE(IUNITO,9180) NFREQ, MAXFRQ
9180     FORMAT(//,' *** ERROR in NASIM: Number of length ',
     &      'classes ',I4,' exceeds program limit of ',I4,'.',/)
         STOP 'NASIM: Dimensions exceeded.'
      END IF
C
C     Initialize the length distribution array.
      DO 10 I=1,NFREQ
         F(I) = 0.
10    CONTINUE
C
C  First generate fish for lengths and then for length-age.
      IF(NFISH .NE. 0) THEN
C
C        Generate age and length of fish.
C        Use ISEEDI as the initial seed for the age and length of
```

```
C        individual fish.
C
         IXO = ISEEDI
C
         SUML = 0.0
         SUMSQR = 0.0
         DO 20 I=1, NFISH
            CALL FGEN( P, NAGE, LAGE, M, S, IL, NFREQ , IXO, L1, L2 ,
     &           L, LINT, NLONG, NSHORT)
C
C           Check if fish lies outside of region of interest.
C
            IF(IL .NE. -1) THEN
C              Accumulate the number in each length category.
C              and the sums of the lengths and lengths squared.
               F(IL) = F(IL) + 1.
               SUML = SUML + L
               SUMSQR = SUMSQR + L*L
            END IF
20       CONTINUE
      END IF
C
C  Find mean and median lengths and std. dev. for the distribution.
C  Also check the total numbers in the distribution.
      NUMFSH = NFISH - NSHORT - NLONG
      LMEAN  = SUML / NUMFSH
      NMEDAN = NUMFSH / 2
      NSUM = 0
      IFREQ = 0
      DO 30 I=1,NFREQ
         NSUM = NSUM + F(I)
         IF(NSUM .GT. NMEDAN .AND. IFREQ .EQ. 0) THEN
C           Found length range in which median falls.
            IFREQ = I
C           Calculate the relative position in the interval.
            PROP = (NMEDAN - (NSUM - F(I)))/F(I)
         END IF
30    CONTINUE
C
C     Set the value of the median to the nearest half interval.
      IF(PROP .LE. .25) THEN
C        Median length is nearest the bottom of the interval.
         FRCINT = 1.0
      ELSE IF(PROP .GT. .75) THEN
C        Median length is near the top of the interval.
         FRCINT = 0.0
      ELSE
C        Median length is nearest the centre of the interval.
         FRCINT = 0.5
      END IF
      LMEDAN = (IFREQ - FRCINT)*LINT + L1
      STDDEV = SQRT( (SUMSQR - (SUML**2)/NUMFSH)/(NUMFSH-1) )
      WRITE(IUNITO,9190)NFREQ, LINT, L1, L2, LMEAN, LMEDAN, STDDEV, NSUM
9190  FORMAT('1',/,' Statistics for the lengths (cm) simulated:',/
     &     ' ---------------------------------------',//,
     &     ' Length categories:',/,
     &     ' Number of categories ',T37,I7,/,
     &     ' Length of interval ',T37,F7.3,/,
     &     ' Length range - lower limit ',T37,F7.3,/,
     &     ' Length range - upper limit ',T37,F7.3,//,
     &     ' For fish within length range:',/,
     &     ' Mean length ',T37, F7.3,/
     &     ' Median length (nearest 1/2 intrvl) ',T37,F7.3,/,
     &     ' Standard deviation in lengths ',T37,F7.3,//,
     &     ' For all fish simulated:',/,
     &     ' Number within length range',',T37,I7)
      IF(NSHORT .NE. 0) WRITE(IUNITO,9200) NSHORT
9200  FORMAT(' Number rejected as too short ',T37,I7)
      IF(NLONG .NE. 0) WRITE(IUNITO,9210) NLONG
9210  FORMAT(' Number rejected as too long  ',T37,I7)
      PFACTR = POPSUM/POPORG
      WRITE(IUNITO,9220) INT(NUMFSH*PFACTR), PFACTR
```

```
9220  FORMAT(' Actual number that are fishable ',T37,I7,/,
     &    ' Proportion that are fishable ',T37,F7.3)
C
      IF(NUMFSH .NE. NSUM) THEN
         STOP 'ERROR: NASIM total of distribution is incorrect'
      END IF
C
      WRITE(INTCRT,9230)
      WRITE(IUNITO,9230)
9230  FORMAT(/,' Simulation is complete.  Distribution follows ...')
      RETURN
C
40    CONTINUE
      WRITE(INTCRT,9240) INFILE
      WRITE(IUNITO,9240) INFILE
9240  FORMAT(/,' *** End of input data in file ',A20,
     &    ' Simulation stopped ***')
      STOP 'NASIM: Unexpected end of file reached on read'
      END
C
C*******************************************************************************
C
      SUBROUTINE NASMCK(L1,L2,NLONG,NSHORT)
C
C     Checks if NASIM generated fish either too long or too short
C     for the lengths of interest.  Prints message if any found.
C
C-------------------------------------------------------------------------------
      REAL L1, L2
      INTEGER NLONG, NSHORT
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
      IF(NLONG .NE. 0) THEN
         WRITE(INTCRT,9000) NLONG, L2
9000     FORMAT(//,' *** Simulation produced ',I5,' fish longer than ',
     &       F6.2,' cm. ***',/,
     &       ' These fish are not included in the distribution',
     &       ' statistics.',//)
         CALL HOLD(INTCRT,INTKEY)
      END IF
      IF(NSHORT .NE. 0) THEN
         WRITE(INTCRT,9010) NSHORT, L1
9010     FORMAT(//,' *** Simulation produced ',I5,' fish shorter than ',
     &       F6.2,' cm. ***',/,
     &       ' These fish are not included in the distribution',
     &       ' statistics.',//)
         CALL HOLD(INTCRT,INTKEY)
      END IF
C
      RETURN
      END
C
C*******************************************************************************
C
      SUBROUTINE PARERR(MAXFRQ, MXFREQ, MAXYRS, MXYRS)
C
C     Subroutine to an print error message when parameter
C     MAXFRQ does not equal variable MXFREQ or when parameter
C     MAXYRS does not equal variable MXYRS.
C
C-------------------------------------------------------------------------------
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
      WRITE(INTCRT,9000) MAXFRQ, MXFREQ, MAXYRS, MXYRS
      WRITE(IUNITO,9000) MAXFRQ, MXFREQ, MAXYRS, MXYRS
9000  FORMAT(//,' **** ERROR in subroutine. ',/,
     &    '       Parameter MAXFRQ = ',I5,
     &    ' not equal to value of ',I10,' in mainprogram.',/,
     &    ' OR ',/,
     &    '       Parameter MAXYRS = ',I5,
```

```
     &    ' not equal to value of ',I10,' in mainprogram.',/)
C
      STOP 'LFMODEL stopped'
      END
C
C*******************************************************************************
C
      FUNCTION RANNUM(IX0)
C
C  A 'dummy' function to provide pseudo-random real numbers in the
C  range from 0.0 to less than 1.0 .
C  IX0 is the seed to be supplied to a random number function
C  subprogram.
C  The seed is updated on return by each functions and used to update
C  IX0.  IX0 should be declared REAL*8 for compatibility with any of
C  the possible functions that could be called below.
C
C  RANNUM allows a choice of the actual function used to provide the
C  pseudo-random numbers.  The seed for the available routines is
C  either an integer or the integer portion of a REAL*8 number.
C  The appropriate variable must be assigned the value of IX0 prior
C  to calling the function subprogram.
C
C  The choice of the function is from the statements below, where the
C  desired routine is active and the rest are commented-out.  This
C  method of choosing a function does not invoke unnecesary libraries,
C  and will likely be done only when using an alternate computer system.
C
C  The present choice is from the following routines:
C
C                  RAN(IXI)     --- a VAX Fortran math library function
C                                   where IXI is an INTEGER*4 seed updated
C                                   by the function.  IXI should be set
C                                   initially to a large, odd-valued
C                                   integer (less than +2 147 483 647)
C
C                  GGUBFS(IXD) --- An International Mathematics and
C                                   Statisics Library (IMSL) function
C                                   where IXD is a REAL*8 seed assigned
C                                   an integer value by the function.
C                                   IXD should be set initially to an
C                                   integer value in the inclusive
C                                   range of 1.D0 to 2 147 483 647.D0 .
C                                   This library routine must be included
C                                   when LINKing via:
C                                   $LINK <program>,sys$SHARE:IMSLIBS/lib
C                                   for PBS, Nanaimo,
C                                   or
C                                   $LINK <program>,ccs$IMSLLIB/lib
C                                   for IOS, Sidney.
C
C                  RAND(RSEED) --- a subroutine obtained from the SLATEC
C                                   public domain library for MS-DOS
C                                   personal computers.
C                                   RSEED is the initial seed or a flag
C                                   to control the random number function
C                                   If RSEED is positive, then the top
C                                   two 11-bits of the fractional part
C                                   of RSEED will be the seed.  If RSEED
C                                   is zero, then the next pseudo-random
C                                   in the series will be generated.
C                                   If RSEED is negative, the last
C                                   generated number will be returned and
C                                   usable as the seed to continue the
C                                   series.
C
C  Note: IX0 should be declared REAL*8 in all routines calling RANNUM.
C
C-------------------------------------------------------------------------------
C
      REAL*8    IX0, IXD, ZERO
      INTEGER*4 IXI
```

```
      LOGICAL FIRST
      DATA FIRST /.TRUE./, ZERO /0.0D0/
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C------- Start of RAN Section - VAX math library routine ---------------
C
C  To use RAN activate this section and comment-out the other two:
C**    IXI    = IXO
C**    RANNUM = RAN(IXI)
C**    IXO    = IXI
C**    IF(FIRST) THEN
C**       FIRST = .FALSE.
C**       WRITE(IUNITO,9000)
C**    END IF
C
C------- End of RAN Section -----------------------------------------------
C
C
C------- Start of GGUBFS Section - IMSL routine ---------------------------
C
C  To use GGUBFS activate this section and comment-out the other two:
C**    IXD    = IXO
C**    RANNUM = GGUBFS(IXD)
C**    IXO    = IXD
C**    IF(FIRST) THEN
C**       FIRST = .FALSE.
C**       WRITE(IUNITO,9010)
C**    END IF
C
C------- End of GGUBFS Section -------------------------------------------
C
C
C------- Start of RAND Section - Local routine ----------------------------
C
C  To use RAND activate this section and comment-out the other two:
C     If the first call, divide seed IXO by 2**22 for initial value.
C     RAND will update the seed internally and maintain its
C     value between calls.
C
C     Check if a new seed has been specified in the calling routine.
      IF(IXO .NE. ZERO) THEN
C        Use a new value for the initial seed for RAND.
         RSEED = IXO/4194304.0
      ELSE
C        Use the next value in the random series.
         RSEED = 0.0
      END IF
      RANNUM = RAND(RSEED)
C
C     Set the value of IXO to zero, so that the next value in the series
C     will be generated unless a non-zero seed is specified in the
C     calling routine.
      IXO = ZERO
      IF(FIRST) THEN
         FIRST = .FALSE.
         WRITE(IUNITO,9020)
      END IF
C
C------- End of RAND Section ----------------------------------------------
C
C
9000  FORMAT(/,' --- Using VAX random number routine RAN ---    ',/)
9010  FORMAT(/,' --- Using IMSL random number routine GGUBFS ---',/)
9020  FORMAT(/,' --- Using local random number function RAND ---',/)
C
      RETURN
      END
C
C*******************************************************************************
```

```
C
      SUBROUTINE SCHNTE(EXPA,B,NAGE,TAUS,TAU1,TAU2,Y1,Y2,Y)
C
CCC   Produces an array of length,Y for NAGE age classes of fish,
CCC   where the growth function calculation is based on the method of
CCC   Jon Schnute as per:
CCC
CCC        Schnute,J. 1981.  A Versatile Growth Model With Statistically
CCC        Stable Parameters.  Canadian Journal of Fisheries and
CCC        Aquatic Sciences. Vol. 38, No. 9, pages 1128-1140.
CCC
CCC   The general form of the growth function is given below:
CCC
CCC   Length at age,t =
CCC          [Y1**B + (Y2**B-Y1**B)*(1.0-EXPA**(t - TAU1))/
CCC                                 (1.0-EXPA**(TAU2-TAU1))]**(1/B)
CCC
CCC   The coefficients used correspond to Schnute's paper, except for
CCC   "EXPA" which equals to the exponential of the negative of his
CCC   coefficient, "a" (i.e. EXPA = EXP(-a).  This change was made to
CCC   reduce the computational time by using A**T, where T is an
CCC   integer interval of time.  (The coding EXP(-a*T) would involve
CCC   a multiplication as well as the power of a noninteger number).
CCC
CCC   The 3 alternate formulations for Schnute's special cases where
CCC   a=0 (i.e. EXPA=1) and/or b=0 are included as well.
CCC
CCC   Calling sequence:-
CCC   Real scalars:
CCC                     EXPA --- first growth coef. (EXPA = EXP(-a),
CCC                              where a is Schnute's first coefficient).
CCC                     B    --- second growth coefficient (same as b,
CCC                              Schnute's second coefficient).
CCC   Integer scalars:
CCC                     NAGE --- number of age classes
CCC                     TAUS --- age of first age class
CCC                     TAU1 --- age of fish of mean length Y1
CCC                     TAU2 --- age of fish of mean length Y2
CCC   Real scalars:
CCC                     Y1   --- mean length of fish at age TAU1
CCC                     Y2   --- mean length of fish at age TAU2
CCC   Real array:
CCC                     Y    --- mean lengths of age classes 1 to NAGE
CCC
CCC   ------------------------------------------------------------------
CCC
CC    Define the minimum difference between coefficients A and zero, B
C     and zero, and B and one:
      PARAMETER (SMALL = 1.0E-6)
C
      INTEGER NAGE, TAUS, TAU1, TAU2, T, TDIFF
      REAL Y(*)
      LOGICAL AZERO, BONE, BZERO
C
CC    Check is coefficients are valid:
      IF(Y2 .LE. Y1 .OR. Y1 .LT. 0.0 .OR. Y2 .LT. 0.0) THEN
         STOP 'SCHNTE: Invalid reference sizes Y1 and/or Y2'
      ELSE IF(TAU2 .LE. TAU1) THEN
         STOP 'SCHNTE: Invalid reference times TAU1 and/or TAU2'
      END IF
C
CC    Determine internal parameters based on the input coefficients:
      IF(ABS(B) .LT. SMALL) THEN
C        Schnute coefficient "b" is zero. (Schnute's cases 2 and 4).
         BZERO = .TRUE.
         BONE  = .FALSE.
         Y1B = 0.0
         Y2B = 0.0
         BINVRS = 0.0
         PART1 = LOG(Y2/Y1)
      ELSE
C        Schnute coefficient "b" is nonzero. (Schnute's cases 1 and 3).
```

```
            BZERO = .FALSE.
            IF(ABS(B - 1.0) .LT. SMALL) THEN
C              Coefficient B is sufficiently near one, so simplify terms.
               BONE = .TRUE.
               Y1B = Y1
               Y2B = Y2
               BINVRS = 1.0
            ELSE
C              General terms for cases 1 and 3.
               BONE = .FALSE.
               Y1B = Y1**B
               Y2B = Y2**B
               BINVRS = 1.0/B
            END IF
            PART1 = Y2B - Y1B
         END IF
C
         IF(ABS(EXPA - 1.0) .LT. SMALL) THEN
C           Schnute coefficient "a" is zero. (Schnute's cases 3 and 4).
C           (since EXPA = EXP(-a), when a=0, then EXPA=1)
            AZERO = .TRUE.
            DENOM = TAU2 - TAU1
         ELSE
C           Schnute coefficient "a" is nonzero. (Schnute's cases 1 and 2).
            AZERO = .FALSE.
            DENOM = 1.0 - EXPA**(TAU2 - TAU1)
         END IF
C
C        Relate the time of each age class to the starting time and allow
C        for the difference between the starting time and the first
C        reference time, TAU1.  Subtract an additional 1, to account
C        for the fact that age classes are counted from I=1 and not I=0.
         TDIFF = (TAUS - TAU1) - 1
C
C
         DO 10 I=1,NAGE
C           Calculate the mean length for each age class.
            T = I + TDIFF
            IF(BZERO) THEN
               IF(AZERO) THEN
C                 Schnute's case 4 formulation (a=0, b=0):
                  Y(I) = Y1 * EXP(PART1 * T / DENOM)
               ELSE
C                 Schnute's case 2 formulation (a .ne. 0, b=0):
                  Y(I) = Y1 * EXP(PART1 * (1.0 - EXPA**T) / DENOM)
               END IF
            ELSE
               IF(AZERO) THEN
C                 Schnute's case 3 formulation (a=0, b .ne. 0):
                  TEMPL = Y1B + (PART1 * T / DENOM)
               ELSE
C                 Schnute's case 1 formulation (a .ne. 0, b .ne. 0):
                  TEMPL = Y1B + (PART1 * (1.0 - EXPA**T) / DENOM)
               END IF
               IF(BONE) THEN
C                 Simply case where b=1:
                  Y(I) = TEMPL
               ELSE
C                 General case.  Result is raised to power of (1/b):
                  Y(I) = TEMPL**BINVRS
               END IF
            END IF
C
10       CONTINUE
C
C
         RETURN
         END
```

APPENDIX J. Program Listing for File MORATE.FOR

The module MORATE to interactively handle mortality schedules.

```
C*************************************************************************
CC                                                                       *
CC     *** Module MORATE ***                                             *
CC                                                                       *
C*************************************************************************
C
      SUBROUTINE MORATE(FYEAR,MAGE,MINAG,MAXAG,MTYPE,NAGE,PREC,SAMORT,
     &   ZFMORT,ZNMORT)
C
CC     An interactive program module to do two functions:
CC          1) Read &/or edit mortality rate schedule
CC          2) Sum annual rates for each age group
CC
CC     The subprogram operates in 3 basic modes:
CC          1) uses the rate schedule exactly as used in the
CC             previous run of the program,
CC          2) uses constant mortality rates for fishing and
CC             natural mortality,
CC          3) uses rates that can vary for each year for fishing
CC             mortality, and each age class for natural mortality.
CC
CC     To use the previous rate schedule, MORATE calls:
CC         MORVRO -- to open and read a file of mortality rates.
CC         MORVWR -- to write a mortality rate schedule.
CC         MORVTR -- to transfer the schedule values to arrays of
CC                   rates for each year/age-class.
CC         MORHLD -- to hold the "screen" before printing any more
CC                   results.
CC         MORSUM -- to calculate the accumulated mortality rates.
CC
CC     To use a constant mortality rates, MORATE calls:
CC         MORCON -- for a constant annual mortality rate.
CC         MORSUM -- to calculate the accumulated mortality rates.
CC     MORCON in turn, calls:
CC         MORCSV -- to save a schedule of constant rates.
CC
CC     To use variable mortality rates, MORATE calls:
CC         MORVAR -- for variable mortality rates.
CC         MORSUM -- to calculate the accumulated mortality rates.
CC     MORVAR, in turn, calls:
CC         MORZRO -- to initialize rate schedule arrays.
CC         MORVRO -- to open and read a file of mortality rates.
CC         MORVED -- to edit a rate schedule.
CC         MORVEX -- to extend a rate schedule.
CC         MORVWR -- to write a mortality rate schedule.
CC         MORVTR -- to transfer the schedule values to arrays of
CC                   rates for each year/age-class.
CC         MORVSV -- to save the editted rate schedule for
CC                   use in a later session.
CC     and  indirectly via MORVED and MORVEX calls:
CC         MORVRR -- to edit the rates in a schedule of rates.
CC         MORVRY -- to edit the ranges of years for the rates
CC                   in a schedule of rates.
CC
C-------------------------------------------------------------------------
CC
CC     Description of functions performed:
CC
CC     1) Mortality rate schedule.
CC        ------------------------
CC     Reads a schedule of annual mortality rates (from units MUNLAS or
CC     MUNSAV) and calculates their sums for each age from MINAG to
CC     MAXAG. If a schedule does not exist, then one is created where
CC     ZFMORT is used for all fishing rates from the present date back
CC     NAGE years and ZNMORT is used for the natural mortality rate for
CC     all age classes.  If a schedule exists, the user has the option
CC     of using it for the simulation, either as is or modified.  If it
CC     is modified, the user also has the option of using the modified
CC     schedule to update the one on unit 11, for future use.
CC     NOTE:-  The schedule must be entered in order of increasing
C      "ancient" years  and age classes.
```

```
C
C     2) Sum of annual rates for each age group.
C     ------------------------------------------
C
C     Using the mortality rate schedule, annual rates are determined
C     for each year covering the age groups of MINAG to MAXAG.  These
C     rates are summed for each group from MINAG to the age of the
C     group.  These sums are returned as elements of the array SAMORT.
C
C  Calling Sequence:
C                FYEAR  --- fishing mortality rate for each year from
C                           the present time back to the time that the
C                           oldest fish was recruited
C                MAGE   --- natural mortality rate for each age class
C                MINAG  --- minimum age in the simulation
C                MAXAG  --- maximum age in the simulation
C                MTYPE  --- 20 character name for type of mortality
C                NAGE   --- number of ages to simulate
C                SAMORT --- array of annual mortality
C                           rates summed over the life of
C                           each age group from MINAG up
C                           to the age of the group.
C                PREC   --- partial recruitment for age classes
C                ZFMORT --- default fishing mortality rate
C                ZNMORT --- default natural mortality rate
C                           Note: If variable rates are to be used
C                           but a rate schedule does not exist
C                           then the value of ZFMORT will be used
C                           for all years back to the year when
C                           the NAGE age class was recruited.
C                           Similarly, ZNMORT will be used for all
C                           age classes up to NAGE.
C
C  Common block /LGUNIT/ variables.
C                INTKEY --- default logical unit for interactive
C                           keyboard input
C                INTCRT --- default logical unit for interactive
C                           display (CRT)
C                IUNITO --- logical unit for output from simulation
C
C
C  Parameters:
C
C                LOGKEY --- default logical unit for interactive input.
C                LOGCRT --- default logical unit for interactive display.
C                MXSCHD --- maximum number of periods in the mortality
C                           rate schedule.
C                MFLLAS --- name of the file containing the default
C                           mortality rate schedule as used for the
C                           LAST run of the program.  (Filename.type
C                           form for VAX, 20 characters maximum).
C                MFLSAV --- name of the file which will contain the
C                           explicitly SAVED mortality rate schedule.
C                           (Filename.type form for VAX, 20 characters
C                           maximum).
C                MUNLAS --- logical unit to use for MFLLAS.
C                MUNSAV --- logical unit to use for MFLSAV.
C
C-----------------------------------------------------------------------
C
      CHARACTER ANSWER*1, MFLLAS*20, MFLSAV*20, IFLLAS*20, IFLSAV*20,
     &   MTYPE*20
C
      PARAMETER (LOGKEY = 5,
     &           LOGCRT = 6,
     &           MXSCHD = 50,
     &           MFLLAS = 'LFMORTRT.LAS        ',
     &           MFLSAV = 'LFMORTRT.SAV        ',
     &           MUNLAS = 11,
     &           MUNSAV = 12 )
C
```

```
      REAL    ARATE(MXSCHD), CRATE(MXSCHD), FYEAR(*), MAGE(*), SAMORT(*)
      INTEGER YEARND(MXSCHD), NDCLAS(MXSCHD), THISYR
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
CC    Set variables equal to the parameter named constants
C     for use between subroutines of this module:
      IF(INTKEY .EQ. 0) INTKEY = LOGKEY
      IF(INTCRT .EQ. 0) INTCRT = LOGCRT
      MAXSCH = MXSCHD
      IFLLAS = MFLLAS
      IFLSAV = MFLSAV
      IUNLAS = MUNLAS
      IUNSAV = MUNSAV
C
C     Check if the mortality rates from the last run are to be used.
10    CONTINUE
      WRITE(INTCRT,9000)
9000  FORMAT(/,' Do you wish to use the mortality rates from the last',
     &    ' run? [y/n] (default=y):',/,
     &    ' If not, then variable rates can be displayed & ',
     &    ' optionally  editted.')
      READ(INTKEY,9010,ERR=10) ANSWER
9010  FORMAT(A1)
C
      IF(ANSWER .NE. 'N' .AND. ANSWER .NE. 'n') THEN
         WRITE(IUNITO,9020)
         WRITE(INTCRT,9020)
9020     FORMAT(/,' --- USING PREVIOUS MORTALITY SCHEDULE ---',/)
C
C        Read rates from last run (default file).
         CALL MORVRO(ARATE,CRATE,MAXSCH,IFLLAS,IUNLAS,NAGE,NDCLAS,
     &      NSCHED,NSCHDF,NSCHDN,YEARND,ZFMORT,ZNMORT,MTYPE)
C
C        Write rate schedule to output unit.
         CALL MORVWR(ARATE,CRATE,IUNITO,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &      YEARND)
C
CC       Transfer the fishing rate schedule values to FYEAR array for
C        each year of the simulation.
         FISHRT = .TRUE.
         CALL MORVIR(FYEAR,.TRUE.,MINAG,MAXAG,NAGE,NSCHDF,ARATE,YEARND)
C
CC       Transfer the natural mortality rate schedule values to MAGE
C        array for each age class in the simulation.
         FISHRT = .FALSE.
         CALL MORVIR(MAGE,.FALSE.,MINAG,MAXAG,NAGE,NSCHDN,CRATE,NDCLAS)
C
C        Wait until user has read screen.
         CALL MORHLD(INTCRT,INTKEY)
      ELSE
C        Can either specify &/or examine previous rate schedules.
C        Check if constant mortality rate is to be used.
20       CONTINUE
         WRITE(INTCRT,9030)
9030     FORMAT(/,' Do you wish to use a constant mortality rate?',
     &      ' [y/n] (default=y):')
         READ(INTKEY,9010,ERR=20) ANSWER
         IF(ANSWER .NE. 'N' .AND. ANSWER .NE. 'n') THEN
C           Use constant mortality rate.
            CALL MORCON(FYEAR,MAGE,IFLLAS,IUNLAS,NAGE,ZFMORT,ZNMORT)
            MTYPE = 'Constant Mortality
         ELSE
C           Use variable mortality rate.
            CALL MORVAR(ARATE,CRATE,FYEAR,IFIRST,MAGE,MAXAG,MAXSCH,
     &         IFLLAS,IFLSAV,MINAG,IUNLAS,IUNSAV,NAGE,NDCLAS,
     &         NSCHED,NSCHDF,NSCHDN,YEARND,ZFMORT,ZNMORT)
            MTYPE = 'Variable Mortality
         END IF
      END IF
C
C     Calculate the accumulated mortality rates for each age class.
```

```
      CALL MORSUM(FYEAR,MAGE,NAGE,PREC,SAMORT)
C
      WRITE(INTCRT,9040)
9040  FORMAT(//,' INTERACTIVE INPUT ENDED.',//,
     &    ' Now simulating the length distribution.',
     &    '  Please wait ...',///)
      RETURN
      END
C
C*******************************************************************************
C
      SUBROUTINE MORHLD(INTCRT,INTKEY)
C
C     Hold screen until user responds with a carriage return.
C     Message written to unit INTCRT. Response is read from unit INTKEY.
C
C-------------------------------------------------------------------------------
C
      CHARACTER RET*1
C
      WRITE(INTCRT,9000)
9000  FORMAT(/,' Press RETURN to continue ...')
      READ(INTKEY,9010) RET
9010  FORMAT(A1)
      RETURN
      END
C
C*******************************************************************************
C
      SUBROUTINE MORCON(FYEAR,MAGE,MFLLAS,MUNLAS,NAGE,ZFMORT,ZNMORT)
C
C     Subroutine to set a constant annual mortality rate for fishing
C     for each year and for natural mortality for each age group.
C     Rate must be positive, and less than or equal to 3.0.
C     If no rate is entered, then the default values are used.
C     Calling sequence variables are defined in MORATE.
C
C-------------------------------------------------------------------------------
C
      CHARACTER MFLLAS*20
      CHARACTER RESPNS*10, BLANK*10
      REAL FYEAR(*), MAGE(*)
      DATA BLANK /'          '/
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
      WRITE(INTCRT,9000)
9000  FORMAT(/,' --- USING CONSTANT MORTALITY ---',/)
C
10    CONTINUE
C     Use a constant fishing mortality rate.
      WRITE(INTCRT,9010) ZFMORT
9010  FORMAT(/,' Enter constant FISHING mortality rate: (default = ',
     &   F5.2,' )',/
     &    ' (1 to 9 digit-decimal number',
     &    ' from 0.0 to 3.0 (format = F10.0))')
      READ(INTKEY,9020,ERR=10) RESPNS
9020  FORMAT(A)
      IF(RESPNS .EQ. BLANK) THEN
C        No value was entered, so use the default of ZFMORT.
         CFRATE = ZFMORT
      ELSE
C        Read the value entered.
         READ(RESPNS,9030,ERR=10) CFRATE
9030     FORMAT(BN,F10.0)
         IF(CFRATE .GT. 3.0 .OR. CFRATE .LT. 0.0) THEN
            WRITE(INTCRT,9040) CFRATE
9040        FORMAT(T5,' Rate entered (',G13.6,') is invalid. ',
     &          ' Please reenter:')
            GO TO 10
         END IF
      END IF
```

```
      WRITE(INTCRT,9050) CFRATE
      WRITE(IUNITO,9050) CFRATE
9050  FORMAT(/,' --- USING CONSTANT FISHING MORTALITY RATE OF ',
     &   F5.2,' ---')
C
C     Use a constant natural mortality rate:
20    CONTINUE
      WRITE(INTCRT,9060) ZNMORT
9060  FORMAT(/,' Enter constant NATURAL mortality rate: (default = ',
     &   F5.2,' )'/
     &   ' (1 to 9 digit-decimal number'
     &   ' from 0.0 to 3.0 (format = F10.0))')
      READ(INTKEY,9020,ERR=10) RESPNS
      IF(RESPNS .EQ. BLANK) THEN
C        No value was entered, so use the default of ZNMORT.
         CNRATE = ZNMORT
      ELSE
C        Read the value entered.
         READ(RESPNS,9030,ERR=20) CNRATE
         IF(CNRATE .GT. 3.0 .OR. CNRATE .LT. 0.0) THEN
            WRITE(INTCRT,9040) CNRATE
            GO TO 20
         END IF
      END IF
      WRITE(INTCRT,9070) CNRATE
      WRITE(IUNITO,9070) CNRATE
9070  FORMAT(/,' --- USING CONSTANT NATURAL MORTALITY RATE OF ',
     &   F5.2,' ---')
C
C     Transfer the constant rates into their rate arrays.
      DO 30 I=1,NAGE
C        Fishing mortality rate for all years back to when the oldest
C        fish was recruited.
         FYEAR(I) = CFRATE
C        Natural mortality rate for all age classes.
         MAGE(I)  = CNRATE
30    CONTINUE
C
C     Save rate schedule:
C
C     A schedule for each run is always written to file MFLLAS,
C     on unit MUNLAS.
C
      CALL MORCSV(CFRATE,CNRATE,MFLLAS,MUNLAS,NAGE)
C
      RETURN
      END
C*******************************************************************************
C
      SUBROUTINE MORCSV(CFRATE,CNRATE,MFLLAS,MUNLAS,NAGE)
C
C     Subroutine to save the constant rate schedule for use in a later
C     session.
C     Calling sequence variables are defined in MORATE.
C
C-------------------------------------------------------------------------------
C
      CHARACTER  MFLLAS*20
      LOGICAL    MEXIST, MOPEN
      REAL       RATEF(1), RATEN(1)
      INTEGER NF(1),     NN(1)
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C     Setup arrays for the write routine.
      RATEF(1) = CFRATE
      RATEN(1) = CNRATE
      NF(1) = NAGE
      NN(1) = NAGE
      NSCHED = 1
      NSCHDF = 1
```

```
      NSCHDN = 1
C
CC    Write to unit MFLLAS to save the current schedule.
C     Open mortality rate schedule backup file.
      INQUIRE(FILE=MFLLAS, OPENED=MOPEN, EXIST=MEXIST, ERR=10)
      IF(.NOT. MOPEN) THEN
C        File is not currently open.
         IF(.NOT. MEXIST) THEN
C           Create a new file.
            OPEN(UNIT=MUNLAS, FILE= MFLLAS, STATUS='NEW', ERR=20)
C           Write rate schedule of fishing and natural mortality rates
C           to unit MUNLAS.
            CALL MORVWR(RATEF,RATEN,MUNLAS,NN,NSCHED,NSCHDF,NSCHDN,NF)
            WRITE(INTCRT,9000) MFLLAS
9000        FORMAT(/,' The current constant rate schedule ',
     &           'has been saved on file ',A20,/)
         ELSE
            OPEN(UNIT=MUNLAS, FILE=MFLLAS , STATUS='OLD', ERR=30)
C           File exists and has not been opened yet.
CC          Write rate schedule of fishing and natural mortality
C           rates to unit MUNLAS.
            CALL MORVWR(RATEF,RATEN,MUNLAS,NN,NSCHED,NSCHDF,NSCHDN,NF)
            WRITE(INTCRT,9000)  MFLLAS
         END IF
C        Close the file containing the last-used schedule.
         CLOSE(UNIT=MUNLAS,STATUS='KEEP')
C
      ELSE
         WRITE(INTCRT,9010) MFLLAS
9010     FORMAT(/,' File : ',A20,' is currently open.  Not updated',/)
         STOP 'MORCSV: ERROR - file already open'
      END IF
C
C
      RETURN
C
10    CONTINUE
      STOP 'MORCSV: Error in inquire on mortality schedule file'
20    CONTINUE
      STOP 'MORCSV: Error in creating mortality schedule file'
30    CONTINUE
      STOP 'MORCSV: Error in opening mortality schedule file'
C
      END
C
CC*****************************************************************************
C
      SUBROUTINE MORSUM(FYEAR,MAGE,NAGE,PREC,SAMORT)

CC    Calculate the accumulated mortality rate for the NAGE age classes.
CC
C-------------------------------------------------------------------------------
CC
C  This subroutine calculates SAMORT(I), the accumulated mortality rate
C  that the I-th age class has experienced since its recruitment age.
C  The accumulated rate includes the individual effects of natural
C  morality, fishing mortality, and partial recruitment, experienced
C  by the I-th age class of fish from its recruitment age to the
C  present time.
C
C  Define SAMORT(I) =  [Sum of PREC(K)*FYEAR(I-K+1)] + [Sum of MAGE(K)]
C                      for K = 1 to I.
C
C            The sum of MAGE(K) for  K=1 to I, is the natural mortality
C            rate experienced by fish of the I-th age class, from their
C            recruitment age up to their present age.
C
C            FYEAR(L) is the fishing mortality rate for (L-1) years ago,
C            where L = 1,2,3....,I and FYEAR(1) is the fishing mortality
C            rate this year, and FYEAR(I) is the rate when the I-th age
C            class was recruited.
```

```
C
C            PREC(K) is the partial recruitment parameter for fish of
C            the K-th age class.
C
C  For example fish in the 5-th age class have survived the following:
C     i) Natural mortality =
C           MAGE(5) for their present age, this year
C         + MAGE(4) for their age, last year
C         + MAGE(3) for their age, the year before that
C         + .....
C         + MAGE(1) for their age, when recruited
C    ii) Fishing mortality =
C           FYEAR(1)*PREC(5) for this year when they are in age class 5
C         + FYEAR(2)*PREC(4) for last year when they were in age class 4
C         + FYEAR(3)*PREC(3) for year before, when they were age class 3
C         + .....
C         + FYEAR(5)*PREC(1) for the year when they were first recruited
C
C-----------------------------------------------------------------------
C
      REAL FYEAR(*), MAGE(*), PREC(*), SAMORT(*)
C
      DO 20 I=1,NAGE
         SUM = 0.0
         IP1 = I + 1
C        Accumulate the mortality rates up to present time for the
C        I-th age class..
         DO 10 K=1,I
            SUM = SUM + PREC(K)*FYEAR(IP1-K) + MAGE(K)
10       CONTINUE
         SAMORT(I) = SUM
20    CONTINUE
C
      RETURN
      END
C
C*****************************************************************************
C
      SUBROUTINE MORVAR(ARATE,CRATE,FYEAR,IFIRST,MAGE,MAXAG,MXSCHD,
     &    MFLLAS,MFLSAV,MINAG,MUNLAS,MUNSAV,NAGE,NDCLAS,
     &    NSCHED,NSCHDF,NSCHDN,YEARND,ZFMORT,ZNMORT)
C
C     Subroutine to set a schedule of varying mortality rate for ranges
C     of years and to transfer schedules to rates for individual years
C     or age classes.
C     Calling sequence variables are defined in MORATE.
C
C-----------------------------------------------------------------------
C
      LOGICAL FISHRT, OPTION
      CHARACTER ANSWER*1, MFLLAS*20, MFLSAV*20, MTYPE*20
      REAL    ARATE(*), CRATE(*), FYEAR(*), MAGE(*)
      INTEGER YEARND(*), NDCLAS(*), THISYR
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C     Variable mortality rate will be used.
C     Initialize schedule values.
      CALL MORZRO(ARATE,CRATE,MXSCHD,NDCLAS,YEARND)
C
      WRITE(INTCRT,9000)
      WRITE(IUNITO,9000)
9000  FORMAT(/,' --- USING VARIABLE MORTALITY ---')
C
      WRITE(INTCRT,9010)
9010  FORMAT(/,' DEFAULT Mortality Schedule:',/)
C
C     Open, read and print the default mortality rate schedule file.
      CALL MORVRO(ARATE,CRATE,MXSCHD,MFLLAS,MUNLAS,NAGE,NDCLAS,
     &    NSCHED,NSCHDF,NSCHDN,YEARND,ZFMORT,ZNMORT,MTYPE)
      ASSIGN 9020 TO IEDFIL
9020  FORMAT(/,' EDITTING DEFAULT MORTALITY SCHEDULE:',/)
```

```
C
10    CONTINUE
      WRITE(INTCRT,9030)
9030  FORMAT(/,' Do you wish to use the above schedule? ',
     &   '[y/n] (default=y):')
      READ(INTKEY,9040,ERR=10) ANSWER
9040  FORMAT(A1)
      IF(ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n') THEN
20       CONTINUE
         WRITE(INTCRT,9050)
9050     FORMAT(/,' Do you wish to see the schedule saved from a ',
     &      'prior run?  [y/n] (default=n)',/,
     &      '(If default or n, then will edit DEFAULT schedule)')
         READ(INTKEY,9040,ERR=20) ANSWER
         IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') THEN
            WRITE(INTCRT,9060)
9060        FORMAT(/,' EXPLICITLY-SAVED Mortality Schedule:',/)
C
C           Open, read and print the mortality rate schedule file
C           saved from a previous run.
            CALL MORVRO(ARATE,CRATE,MXSCHD,MFLSAV,MUNSAV,NAGE,NDCLAS,
     &         NSCHED,NSCHDF,NSCHDN,YEARND,ZFMORT,ZNMORT,MTYPE)
            ASSIGN 9070 TO IEDFIL
9070        FORMAT(/,' EDITTING EXPLICITLY-SAVED MORTALITY SCHEDULE:',/)
C
30          CONTINUE
            WRITE(INTCRT,9080)
9080        FORMAT(/,' Do you wish to use the above schedule? ',
     &         '[y/n] (default=y):',/
     &         '(If n, then will edit EXPLICITLY-SAVED schedule)')
C
C           Check if this schedule is to used without changes.
            READ(INTKEY,9040,ERR=30) ANSWER
            IF(ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n') THEN
C              Edit the schedule saved from a previous run.
            ELSE
C              Use the schedule saved from a previous run.
               GO TO 80
            END IF
         END IF
C
CC       Edit mortality schedule:
CC
C        Write message to indicate which file is being editted.
         WRITE(INTCRT,IEDFIL)
C
C        Determine which rates to edit: fishing &/or natural mortality.
40       CONTINUE
         WRITE(INTCRT,9090)
9090     FORMAT(//,' Do you wish to edit the FISHING mortalities? ',
     &      '[y/n] (default=n):')
         READ(INTKEY,9040,ERR=40) ANSWER
         IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') THEN
            CALL MORVED(IFIRST,MXSCHD,NS,NSCHDF,ARATE,YEARND)
C
CC          Extend this schedule?
50          CONTINUE
            WRITE(INTCRT,9100)
9100        FORMAT(//,' Do you wish to EXTEND this schedule? ',
     &         '[y/n] (default=n): ')
            READ(INTKEY,9040,ERR=50) ANSWER
            IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') THEN
C
CC             Extend mortality schedule:
               WRITE(INTCRT,9110)
9110           FORMAT(/,' EXTENDING FISHING MORTALITY SCHEDULE:',/)
C
               CALL MORVEX(IFIRST,MXSCHD,NAGE,NS,NSCHDF,ARATE,YEARND)
C
CC             Loop back to extend schedule to another rate period.
               GO TO 50
            END IF
```

```
            END IF
C
60          CONTINUE
            WRITE(INTCRT,9120)
9120        FORMAT(//,' Do you wish to edit the NATURAL mortalities? ',
     &           '[y/n] (default=n):')
            READ(INTKEY,9040,ERR=60) ANSWER
            IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') THEN
               CALL MORVED(IFIRST,MXSCHD,NS,NSCHDN,CRATE,NDCLAS)
C
C              Extend this schedule?
70             CONTINUE
               WRITE(INTCRT,9100)
               READ(INTKEY,9040,ERR=70) ANSWER
               IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y') THEN
C
C                 Extend mortality schedule:
                  WRITE(INTCRT,9130)
9130              FORMAT(/,' EXTENDING NATURAL MORTALITY SCHEDULE:',/)
C
                  CALL MORVEX(IFIRST,MXSCHD,NAGE,NS,NSCHDN,CRATE,NDCLAS)
C
C                 Loop back to extend schedule to another rate period.
                  GO TO 70
               END IF
            END IF
C
CC          All editting and extending of schedules is complete.
CC          Reset NSCHED to be equal to the length of the longer
C           of either the fishing or natural rate schedules.
            NSCHED = MAX(NSCHDF,NSCHDN)
         END IF
C
CC
80       CONTINUE
C        Write mortality schedule to unit 6 for verification.
         WRITE(INTCRT,9140)
9140     FORMAT(/,' FINAL Mortality Schedule:',/)
         WRITE(IUNITO,9150)
9150     FORMAT(/,' Mortality Rate Schedule in the Simulation:',/)
C        Write rate schedule of fishing and natural mortality rates
C        to units INTCRT and IUNITO.
         CALL MORVWR(ARATE,CRATE,INTCRT,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &      YEARND)
         CALL MORVWR(ARATE,CRATE,IUNITO,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &      YEARND)
C
CC       Transfer the fishing rate schedule values to FYEAR array for
C        each year of the simulation.
         FISHRT = .TRUE.
         CALL MORVTR(FYEAR,FISHRT,MINAG,MAXAG,NAGE,NSCHDF,ARATE,YEARND)
C        Transfer the natural mortality rate schedule values to MAGE
C        array for each age class in the simulation.
         FISHRT = .FALSE.
         CALL MORVTR(MAGE,FISHRT,MINAG,MAXAG,NAGE,NSCHDN,CRATE,NDCLAS)
C
CC       Save rate schedules:
CC
CC       1) A schedule for each run is always written to file MFLLAS,
CC       on unit MUNLAS.  (OPTION = .FALSE.)
C
         OPTION = .FALSE.
         CALL MORVSV(ARATE,CRATE,MFLLAS,MUNLAS,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &      YEARND,OPTION)
C
CC       2) Optionally, the rate schedule is written to file MFLSAV, on
CC       unit MUNSAV.  (OPTION = .TRUE.)
CC       If this file already exists, then the user must explicitly answer
CC       yes, when asked to if the existing file's schedule is be replaced
CC       by the unit MUNSAV.
CC       If the file does not exist, one is created, and the current
C        schedule is written into it, without asking the user.
```

```
C
      OPTION = .TRUE.
      CALL MORVSV(ARATE,CRATE,MFLSAV,MUNSAV,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &     YEARND,OPTION)
C
      RETURN
      END
C*****************************************************************************
C
      SUBROUTINE MORVED(IFIRST,MXSCHD,NS,NUMVAL,RATE,YEAR)
C
CC    Edit mortality schedule.
CC    Calling sequence variables are defined in MORATE.
CC
C-----------------------------------------------------------------------------
C
      REAL    RATE(*)
      INTEGER YEAR(*), THISYR
      CHARACTER ANSWER*1
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
CC       Edit mortality schedule.
CC       Set initial values for the starting year for the range that the
C        NS-th rate applies.
      IFIRST = 1
      NS = 1
C
10    CONTINUE
      THISYR = YEAR(NS)
      IF(THISYR .LT. IFIRST) THEN
         THISYR = IFIRST
         YEAR(NS) = THISYR
      END IF
      WRITE(INTCRT,9000) NS, RATE(NS), IFIRST, THISYR
9000  FORMAT(/,' Schedule item ',I3,' :',/,
     &   T5,' Rate of ',F5.2,' will apply from years/age-classes ',
     &   I3,' to ',I3,' inclusive.',/)
      WRITE(INTCRT,9010)
9010  FORMAT(T5,' Is this okay? [y/n] (default=y): ')
      READ(INTKEY,9020,ERR=10) ANSWER
9020  FORMAT(A1)
      IF(ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n') THEN
C        Change the NS-th rate?
         CALL MORVRR(RATE,NS)
C
CC          And/or change the range of years for this rate?
         CALL MORVRY(RATE,IFIRST,NS,NUMVAL,THISYR,YEAR)
C
         WRITE(INTCRT,9030) NS, RATE(NS), IFIRST, YEAR(NS)
9030     FORMAT(/,' Revised item ',I3,')',/,T5,
     &        ' Rate of ',F5.2,' will apply from years/age-classes ',
     &      I3,' to ',I3,' inclusive.',/)
      END IF
      IF(NS .LT. NUMVAL .AND. NUMVAL .LT. MXSCHD) THEN
         IFIRST = YEAR(NS) + 1
         NS = NS + 1
C           Loop back to edit next rate period.
         GO TO 10
      END IF
C
      RETURN
      END
C*****************************************************************************
C
      SUBROUTINE MORVEX(IFIRST,MXSCHD,NAGE,NS,NUMVAL,RATE,YEAR)
C
CC    Extending the rate schedule.
CC    Calling sequence variables are defined in MORATE.
C
```

```
C-----------------------------------------------------------------------
      REAL    RATE(*)
      INTEGER YEAR(*), THISYR
      CHARACTER ANSWER*1
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
10    CONTINUE
      IF(NS .LT. MXSCHD) THEN
C             Set defaults:
         NUMVAL = NUMVAL + 1
         IFIRST = YEAR(NS) + 1
         RATE(NS+1) = RATE(NS)
         NS = NS + 1
         YEAR(NS) = NAGE
         THISYR = YEAR(NS)
         IF(THISYR .LT. IFIRST) THEN
            THISYR = IFIRST
            YEAR(NS) = THISYR
         END IF
20       CONTINUE
         WRITE(INTCRT,9000) NS, RATE(NS), IFIRST, THISYR
9000     FORMAT(/,' Schedule item ',I3,' :',/,T5,' Rate of ',
     &      F5.2,' will apply from years/age-classes ',I3,
     &      ' to ',I3,' inclusive.',/)
         WRITE(INTCRT,9010)
9010     FORMAT(T5,' Is this okay? [y/n] (default=y): ')
         READ(INTKEY,9020,ERR=20) ANSWER
9020     FORMAT(A1)
         IF(ANSWER .EQ. 'N' .OR. ANSWER .EQ. 'n') THEN
C                Change the NS-th rate?
            CALL MORVRR(RATE,NS)
C
C                And/or change the range of years for this rate?
            CALL MORVRY(RATE,IFIRST,NS,NUMVAL,THISYR,YEAR)
C
            WRITE(INTCRT,9030) NS, RATE(NS), IFIRST, YEAR(NS)
9030        FORMAT(/,' Revised item   ',I3,')',/,T5,' Rate of ',
     &         F5.2,' will apply from years/age-classes ',I3,
     &         ' to ',I3,' inclusive.',/)
         END IF
C
      END IF
C
      RETURN
      END
C
C**********************************************************************
C
      SUBROUTINE MORVRO(ARATE,CRATE,MXSCHD,MFLLAS,MUNLAS,NAGE,NDCLAS,
     &   NSCHED,NSCHDF,NSCHDN,YEARND,ZFMORT,ZNMORT,MTYPE)
C
C     Subroutine to open and read a file of varying mortality rates.
C
C-----------------------------------------------------------------------
C
C     Calling sequence:
C
C     ARATE  --- rates for fishing mortality rate schedule
C     CRATE  --- rates for natural mortality rate schedule
C     MXSCHD --- max. no. of schedule items permitted
C     MFLLAS --- name of the file to open
C     MUNLAS --- unit number of the file to open
C     NAGE   --- number of age classes
C     NDCLAS --- ending age classes for the natural rate schedule
C     NSCHED --- the larger of the number of schedule items for either
C                fishing mortality or natural mortality
C     NSCHDF --- number of nonzero schedule items for fishing mortality
C     NSCHDN --- number of nonzero schedule items for natural mortality
C     YEARND --- ending years for the fishing schedule items
C     ZFMORT --- default fishing mortality rate
C     ZNMORT --- default natural mortality rate
```

```
C
C-----------------------------------------------------------------------
C
C     Set default ending year for a rate schedule.
      INTEGER DEFLTY
      PARAMETER (DEFLTY = 150)
C
      CHARACTER MFLLAS*20, MTYPE*20
      REAL      ARATE(*), CRATE(*)
      INTEGER YEARND(*), NDCLAS(*)
      LOGICAL  MEXIST, MOPEN
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C     Open mortality rate schedule file.
      INQUIRE(FILE=MFLLAS, OPENED=MOPEN, EXIST=MEXIST, ERR=70)
      IF(.NOT. MOPEN) THEN
C        File is not currently open.
         IF(.NOT. MEXIST) THEN
C           Create a new file.
            OPEN(UNIT=MUNLAS, FILE= MFLLAS, STATUS='NEW', ERR=80)
C           Assign default mortality schedule where ZFMORT and
C           ZNMORT are used for all age groups for this run.
            NSCHED = 1
            NSCHDF = 1
            NSCHDN = 1
            ARATE(1) = ZFMORT
            CRATE(1) = ZNMORT
            YEARND(1) = MAX(NAGE, DEFLTY)
            NDCLAS(1) = MAX(NAGE, DEFLTY)
C
C           Write rate schedule of fishing and natural mortality rates
C           to unit MUNLAS.
            CALL MORVWR(ARATE,CRATE,MUNLAS,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &         YEARND)
         ELSE
C           File exists and has not been opened yet.
            OPEN(UNIT=MUNLAS, FILE=MFLLAS , STATUS='OLD', ERR=90)
C           Read mortality schedule from file.
C           (First, skip of 4 line-heading)
            READ(MUNLAS,9000)
9000        FORMAT(///)
            DO 10 NS=1,MXSCHD
               READ(MUNLAS,9010,END=20) ARATE(NS), YEARND(NS),
     &            CRATE(NS), NDCLAS(NS)
9010           FORMAT(F10.2,T24,I3,T41,F10.2,T64,I3)
               NSCHED = NS
10          CONTINUE
C           Maximum number of schedule items read.
            WRITE(IUNITO,9020) MXSCHD
            WRITE(INTCRT,9020) MXSCHD
9020        FORMAT(/,' The limit of ',I3,' items read from the ',
     &         'mortality schedule.',/
     &         ' Change MXSCHD in MORATE if more required')
C
C
20          CONTINUE
C
C           Determine number of schedule items for each mortality type.
            DO 30 NS=1,NSCHED
               IF(YEARND(NS) .EQ. 0) THEN
C                 Year is zero, fishing mortality schedule has ended.
                  NSCHDF = NS - 1
                  GO TO 40
               END IF
30          CONTINUE
C           Fishing mortality has NSCHED items.
            NSCHDF = NSCHED
40          CONTINUE
C
            DO 50 NS=1,NSCHED
               IF(NDCLAS(NS) .EQ. 0) THEN
```

```
C                 Age class is zero.  Natural mortality schedule end.
                  NSCHDN = NS - 1
                  GO TO 60
               END IF
50          CONTINUE
C           Natural mortality has NSCHED items.
            NSCHDN = NSCHED
60          CONTINUE
         END IF
      END IF
C
CC    Write rate schedule of fishing and natural mortality rates
C     to unit INTCRT for verification.
      CALL MORVWR(ARATE,CRATE,INTCRT,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &   YEARND)
C
C     Set MTYPE as a message to show the type of mortality schedule.
      IF(NSCHED .EQ. 1) THEN
C        Only one rate for all years and one for all ages.
         MTYPE = 'Constant Mortality '
      ELSE
         MTYPE = 'Variable Mortality  '
      END IF
C
C     Close unit containing the rate schedule.
      CLOSE(UNIT=MUNLAS,STATUS='KEEP')
      RETURN
C
70    CONTINUE
      STOP 'MORVRO: Error in inquire on mortality schedule file'
80    CONTINUE
      STOP 'MORVRO: Error in creating mortality schedule file'
90    CONTINUE
      STOP 'MORVRO: Error in opening mortality schedule file'
C
      END
C
C*******************************************************************************
C
      SUBROUTINE MORVRR(RATE,NS)
C
C     Subroutine to edit the rates in a schedule of rates.
C     Rates must be positive, and less than or equal to 3.0.
C     If no rate is entered, then the value of RATE(NS) is unchanged.
C
C-------------------------------------------------------------------------------
C
      CHARACTER RESPNS*10, BLANK*10
      REAL RATE(*)
      DATA BLANK /'          '/
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C
C     Write present rate value:
10    CONTINUE
      WRITE(INTCRT,9000) RATE(NS)
9000  FORMAT(T5,' Present rate is ',F5.2,
     &   '    Enter new rate? : (default: no change)',/,
     &   T5,' (1 to 9 digit-decimal number',
     &   ' from 0.0 to 3.0 (format = F10.0))')
      READ(INTKEY,9010,ERR=10) RESPNS
9010  FORMAT(A)
      IF(RESPNS .EQ. BLANK) THEN
C        No value was entered, so leave the RATE(NS) as it was.
         WRITE(INTCRT,9020) RATE(NS)
9020     FORMAT(T5,' Rate remains at ',F5.2,'.')
      ELSE
C        Read the value entered.
         READ(RESPNS,9030,ERR=10) RATENS
9030     FORMAT(BN,F10.0)
         IF(RATENS .GT. 3.0 .OR. RATENS .LT. 0.0) THEN
```

```
            WRITE(INTCRT,9040) RATENS
9040        FORMAT(T5,' Rate entered (',G13.6,') is invalid. ',
     &           ' Please reenter:')
            GO TO 10
         ELSE
            WRITE(INTCRT,9050) RATE(NS), RATENS
9050        FORMAT(T5,' Previous rate of ',F5.2,
     &           ' has been replaced by ',F5.2,'.'/)
            RATE(NS) = RATENS
         END IF
      END IF
C
      RETURN
      END
C
C*****************************************************************************
C
      SUBROUTINE MORVRY(RATE,IFIRST,NS,NUMVAL,THISYR,YEAR)
C
C     Subroutine to edit the year-ranges,YEAR for the rates,RATE in a
C     schedule of NUMVAL mortality rates. Edit the NS-th rate applied
C     from year IFIRST to THISYR.
C
C-----------------------------------------------------------------------------
C
      REAL    RATE(*)
      INTEGER YEAR(*), THISYR
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
C
C     Write the range of years for NS-th rate.
10    CONTINUE
      WRITE(INTCRT,9000) IFIRST, THISYR, THISYR
9000  FORMAT(T5,' It will apply from years/age-classes ',I3,
     &    ' to ',I3,' inclusive.',//,
     &    T5,' Change last year/age-class (',I3,
     &    ')?  Enter new value: (default: no change)')
20    CONTINUE
      READ(INTKEY,9010,ERR=10) NEWYR
9010  FORMAT(I10)
      IF(NEWYR .NE. 0) THEN
C
C        Check the year range.
         IF(NEWYR .LT. IFIRST) THEN
            WRITE(INTCRT,9020) IFIRST
9020        FORMAT(T5,' Value must be .ge. ',I3,
     &           ' Please reenter:')
            GO TO 20
         END IF
C        Check if all items of the original schedule
C        have been examined.
         IF(NS .EQ. NUMVAL) THEN
            YEAR(NS) = NEWYR
C              Default schedule editing is complete.
C              Can add new rates periods.
         ELSE
C              Check if NEWYR will replace subsequent periods
C              in the rate schedule.
            NSP1 = NS + 1
            IF(NEWYR .LT. YEAR(NSP1)) THEN
C                 Period is valid, and does not remove all of
C                 the next period in the rate schedule.
               YEAR(NS) = NEWYR
            ELSE
C                 New period will eliminate some items from
C                 the previous version of the rate schedule.
               YEAR(NS) = NEWYR
C                 Check how many items to remove.
               NOUT = 0
               DO 30 IS=NSP1,NUMVAL
                  IF(NEWYR .GE. YEAR(IS)) THEN
```

```
                  NOUT = NOUT + 1
               ELSE
C
C                 Year end for IS-th period is greater
                  than NEWYR, so stop searching.
                  GO TO 40
               END IF
30          CONTINUE
C
40          CONTINUE
C              Remove the first NOUT items after NEWYR.
            IF(NSP1 .EQ. NUMVAL) THEN
               NUMVAL = NS
            ELSE
C
C              Shift schedule items so first valid period
               starts after NEWYR at item NS.
               IPOS = NSP1
               DO 50 IS=NSP1+NOUT,NUMVAL
                  RATE(IPOS) = RATE(IS)
                  YEAR(IPOS) = YEAR(IS)
                  IPOS = IPOS + 1
50             CONTINUE
               NUMVAL = NUMVAL - NOUT
            END IF
         END IF
      END IF
   END IF
   RETURN
   END
C
C*******************************************************************************
C
      SUBROUTINE MORVSV(ARATE,CRATE,MFLSAV,MUNSAV,NDCLAS,NSCHED,NSCHDF,
     &   NSCHDN,YEARND,OPTION)
C
CCC   Subroutine to save the editted rate schedule for use in a later
CCC   session.
CCC   See MORVRO for definition of the calling sequence.
CC----------------------------------------------------------------------
CCC   Save rate schedules:
CCC
CCC   If this file already exists, or OPTION is true, then the user
CCC   must explicitly answer yes, when asked to if the existing
CCC   file's schedule is be replaced by the unit MUNSAV.
CCC   If the file does not exist, one is created, and the current
CCC   schedule is written into it, without asking the user.
CC----------------------------------------------------------------------
C
      CHARACTER ANSWER*1, MFLSAV*20
      REAL    ARATE(*), CRATE(*)
      INTEGER YEARND(*), NDCLAS(*)
      LOGICAL MEXIST, MOPEN, OPTION
C
      COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
CCC   Write to unit MFLSAV to save the current schedule.
C     Open mortality rate schedule backup file.
      INQUIRE(FILE=MFLSAV, OPENED=MOPEN, EXIST=MEXIST, ERR=10)
      IF(.NOT. MOPEN) THEN
C        File is not currently open.
C        IF(.NOT. MEXIST) THEN
C           Create a new file.
            OPEN(UNIT=MUNSAV, FILE= MFLSAV, STATUS='NEW', ERR=20)
CC          Write rate schedule of fishing and natural mortality rates
C           to unit MUNSAV.
            CALL MORVWR(ARATE,CRATE,MUNSAV,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &         YEARND)
            WRITE(INTCRT,9000) NSCHED, MFLSAV
9000        FORMAT(/,' The latest rate schedule (',I3,' item(s)) ',
     &         'has been saved on file ',A20,/)
```

```
            ELSE
               OPEN(UNIT=MUNSAV, FILE=MFLSAV , STATUS='OLD', ERR=30)
C              File exists and has not been opened yet.
               IF(OPTION) THEN
                  WRITE(INTCRT,9010)
9010              FORMAT(//,' **** An EXPLICITLY-SAVED rate schedule',
     &                 ' already exists. ****',/
     &                 ' Do you wish to replace it with the above ',
     &                 'schedule? [y/n] (default=n): ')
                  READ(INTKEY,9020) ANSWER
9020              FORMAT(A1)
               END IF
               IF(ANSWER .EQ. 'Y' .OR. ANSWER .EQ. 'y' .OR.
     &            .NOT. OPTION) THEN
C                 Write rate schedule of fishing and natural mortality
C                 rates to unit MUNSAV.
                  CALL MORVWR(ARATE,CRATE,MUNSAV,NDCLAS,NSCHED,NSCHDF,
     &               NSCHDN,YEARND)
                  WRITE(INTCRT,9000) NSCHED, MFLSAV
               END IF
            END IF
C           Close the file containing the last-used schedule.
            CLOSE(UNIT=MUNSAV,STATUS='KEEP')
C
         ELSE
            WRITE(INTCRT,9030) MFLSAV
9030        FORMAT(/,' File : ',A20,' is currently open.  Not updated',/)
            STOP 'MORVSV: ERROR - file already open'
         END IF
C
C
         RETURN
C
10       CONTINUE
         STOP 'MORVSV: Error in inquire on mortality schedule file'
20       CONTINUE
         STOP 'MORVSV: Error in creating mortality schedule file'
30       CONTINUE
         STOP 'MORVSV: Error in opening mortality schedule file'
C
         END
C
C*************************************************************************
C
         SUBROUTINE MORVIR(ARRAY,FISHRT,MINAG,MAXAG,NAGE,NUMVAL,RATE,YEAR)
C
C        Subroutine to transfer rate schedule arrays (RATE and YEAR values)
C        to a rate array (ARRAY) for each year or age class.
C
C        If FISHRT is .true., any warning messages form this routine will
C        apply to fishing rates.  If it is false, then, natural mortality
C        rates are assumed.
C
C        For fishing rates, ARRAY is equivalent to FYEAR, RATE to ARATE,
C        and YEAR to YEARND.
C
C        For natural mortality rates, ARRAY is equivalent to MAGE, RATE to
C        CRATE, and YEAR to NDCLAS.
C
C        MINAG  --- age of the first age class (age of recruitment)
C        MAXAG  --- age of the oldest age class
C        NAGE   --- number of years and age class to simulate
C        NUMVAL --- number of nonzero mortality values in schedule
C
C-------------------------------------------------------------------------
C
         LOGICAL FISHRT
         REAL    ARRAY(*), RATE(*)
         INTEGER YEAR(*)
C
         COMMON /LGUNIT/ INTKEY, INTCRT, IUNITO
C
```

```
C
C     Define variable equal to 1 less than the minimum age.
      MINAG1 = MINAG - 1
C
CC    Last nonzero year and rate values.
      LASTYR = YEAR(NUMVAL)
      RATLST = RATE(NUMVAL)
C
      ISHORT = 0
      DO 30 I=1,NAGE
C        Find rates that have been experienced by the I-th group of
CC       fish for the last I-years.  All fish in this group are at
CC       age II = I + MINAG - 1, and have been available for fishing
CC       since the age of MINAG.
         IF(I .LE. LASTYR) THEN
C           I-th age group is covered by schedule.
C           Find the rate for this age group.
            IFIRST = 0
            DO 10 NS=1,NUMVAL
               IF(I .GE. IFIRST .AND. I .LE. YEAR(NS)) THEN
C                 Rate for age group has been found.
                  RATENS = RATE(NS)
                  GO TO 20
               END IF
               IFIRST = YEAR(NS)
10          CONTINUE
C
20          CONTINUE
         ELSE
C           Mortality schedule was too short for the range of years
CC          or ages specified.  Use the last nonzero rate in the
C           schedule.
            RATENS = RATLST
            IF(ISHORT .EQ. 0) THEN
               IF(FISHRT) THEN
C                 Fishing mortality rate schedule was short.
                  ISHORT = I
               ELSE
C                 Natural mortality schedule was short.
                  ISHORT = I + MINAG1
               END IF
            END IF
         END IF
C
C        Rate value for this year or age class has been found.
         ARRAY(I) = RATENS
30    CONTINUE
C
      IF(ISHORT .NE. 0) THEN
         IF(FISHRT) THEN
C           Fishing mortality schedule was too short.
            WRITE(INTORT,9000) RATLST, ISHORT, MAXAG
            WRITE(IUNITO,9000) RATLST, ISHORT, MAXAG
         ELSE
C           Natural mortality schedule was too short.
            WRITE(INTORT,9010) RATLST, ISHORT, MAXAG
            WRITE(IUNITO,9010) RATLST, ISHORT, MAXAG
         END IF
9000     FORMAT(/,' *** Specified FISHING mortality schedule was too ',
     &        'short.  '
     &        ,/,' Rate of ',F5.2,' assumed for period of from ',I3,
     &        ' to ',I3,' years ago. ',/)
9010     FORMAT(/,' *** Specified NATURAL mortality schedule was too ',
     &        'short.  '
     &        ,/,' Rate of ',F5.2,' assumed for fish from ',I3,
     &        ' to ',I3,' years old. ',/)
      END IF
C
      RETURN
      END
C
C*****************************************************************************
```

```
C
      SUBROUTINE  MORZRO(ARATE,CRATE,MXSCHD,NDCLAS,YEARND)
C
CCC   Sets all mortality rate schedule items to zero.
CCC
CCC   Calling sequence:
CCC
CCC   Real arrays:
CCC               ARATE(K)   --- annual fishing mortality rate for years
CCC                              after YEARND(K-1) to YEARND(I).
CCC                              If K=1, the rate applies from the
CCC                              present time back to YEARND(K) years ago.
CCC               CRATE(I)   --- natural mortality rate for age class
CCC                              NDCLAS(I) to, but not including
CCC                              NDCLAS(I-1).
CCC                              If I=1, age classes start from 1.
CCC   Integer scalar:
CCC               MXSCHD     --- maximum number of items allowed
CCC                              in the mortality schedule.
CCC   Integer arrays:
CCC               NDCLAS(I)  --- oldest age class to which the natural
CCC                              mortality rate CRATE(I) applies.
CCC               YEARND(I)  --- most "ancient" year to which the fishing
CCC                              mortality rate ARATE(I) applies.
CCC-----------------------------------------------------------------------
C
      REAL    ARATE(*),  CRATE(*)
      INTEGER YEARND(*), NDCLAS(*), MXSCHD
C
      DO 10 I=1,MXSCHD
         ARATE(I) = 0.0
         CRATE(I) = 0.0
         YEARND(I) = 0
         NDCLAS(I) = 0
10    CONTINUE
C
      RETURN
      END
C*****************************************************************************
C
      SUBROUTINE MORVWR(ARATE,CRATE,IUNIT,NDCLAS,NSCHED,NSCHDF,NSCHDN,
     &   YEARND)
C
CCC    Write rate schedule of fishing and natural mortality rates
CCC    to unit IUNIT.
CCC    See MORVRO for definition of the calling sequence.
CCC-----------------------------------------------------------------------
C
      REAL    ARATE(*), CRATE(*)
      INTEGER YEARND(*), NDCLAS(*)
C
      WRITE(IUNIT,9000)
9000  FORMAT(' FISHING Mortality Rate Schedule:',
     &   T42,'NATURAL Mortality Rate Schedule:',//,
     &   ' Mortality Rate',T18,'  Year rate ends  ',
     &   T42,'Mortality Rate',T61,'Rate ends in',/,
     &   '    for period  ',T18,'(backward in time)',
     &   T42,'   for classes',T61,'age class:')
C
CC    Print portion of schedule where both fishing and
C     natural mortality schedules have valid values.
      NBOTH = MIN(NSCHDF,NSCHDN)
      DO 10 NS=1,NBOTH
         WRITE(IUNIT,9010) ARATE(NS), YEARND(NS), CRATE(NS), NDCLAS(NS)
9010     FORMAT(F10.2,T24,I3,T41,F10.2,T64,I3)
10    CONTINUE
C
C
      IF(NSCHDF .EQ. NSCHDN) THEN
```

```
C        Printing is complete.
         RETURN
      ELSE
C        Still more to print.
         IF(NSCHDF .EQ. NSCHED) THEN
C           Only fishing mortalities left to print.
            DO 20 NS=NBOTH+1,NSCHED
               WRITE(IUNIT,9010) ARATE(NS), YEARND(NS)
20          CONTINUE
         ELSE
C           Only natural mortalities left to print.
            DO 30 NS=NBOTH+1,NSCHED
               WRITE(IUNIT,9020) CRATE(NS), NDCLAS(NS)
30          CONTINUE
9020        FORMAT(T41,F10.2,T64,I3)
         END IF
      END IF
C
      RETURN
      END
```

APPENDIX K. Program listing for File RAND.FOR

A function to generate uniformly
distributed pseudo-random numbers.

```
C***  RAND  ************************************************************
C
      FUNCTION RAND (R)
C
C***  Begin prologue  RAND
C
C-----------------------------------------------------------------------
C
C    Obtained from the public domain SLATEC library.
C    Revised to F77, upper/lowercased comments, additional comments and
C    other cosmetics by M.C. Rasmussen, 1988-01-06.
C    Also added a SAVE statement for IX0, IX1, the internal values of the
C    seeds, and a PARAMETER statement for constants IA1, IA0 and IC
C    instead of a DATA statement.
C    A logical parameter (NOPRNT) was added to control the printing
C    of a message the first time this routine is accessed.  If NOPRNT
C    is .TRUE., the message will not be printed.
C
C    Last revision by M.C. Rasmussen, 1988-03-09.
C    Explicitly declared all of the variables used in this program.
C    Integers were declared as INTEGER*4, to require 32-bit integers
C    to be used, even on a 16-bit computer.  This will prevent
C    integer overflow on microcomputers that support the INTEGER*4
C    declaration.
C
C-----------------------------------------------------------------------
C***  Revision date  811015  (yymmdd)
C***  Category no.  G5E
C***  Keywords: random number, uniform, special function
C***  Date written  April 1977
C***  Author  Fullerton W. (LASL)
C***
C***  Purpose:
C             Generates a uniformly distributed random number.
C
C***  Description:
C April 1977 version.  W. Fullerton, C3, Los Alamos Scientific Lab.
C
C     This pseudo-random number generator is portable among a wide
C variety of computers.  RAND(R) undoubtedly is not as good as many
C readily available installation dependent versions, and so this
C routine is not recommended for widespread usage.  Its redeeming
C feature is that the exact same random numbers (to within final round-
C off error) can be generated from machine to machine.  Thus, programs
C that make use of random numbers can be easily transported to and
C checked in a new environment.
C     The random numbers are generated by the linear congruential
C method described, e.g., by Knuth in Seminumerical Methods (p.9),
C Addison-Wesley, 1969.  Given the I-th number of a pseudo-random
C sequence, the I+1 -st number is generated from
C
C                  X(I+1) = (A*X(I) + C) MOD M,
C
C where here M = 2**22 = 4194304, C = 1731 and several suitable values
C of the multiplier A are discussed below.  Both the multiplier A and
C random number X are represented in double precision as two 11-bit
C words.  The constants are chosen so that the period is the maximum
C possible, 4194304.
C
C     In order that the same numbers be generated from machine to
C machine, it is necessary that 23-bit integers be reducible modulo
C 2**11 exactly, that 23-bit integers be added exactly, and that 11-bit
C integers be multiplied exactly.  Furthermore, if the restart option
C is used (where R is between 0 and 1), then the product R*2**22 =
C R*4194304 must be correct to the nearest integer.
C
C     The first four random numbers should be .0004127026,
C .6750836372, .1614754200, and .9086198807.  The tenth random number
C is .5527787209, and the hundredth is .3600893021 .  The thousandth
C number should be .2176990509 .
C
C     In order to generate several effectively independent sequences
```

```
C with the same generator, it is necessary to know the random number
C for several widely spaced calls.  The I-th random number times 2**22,
C where I = K*P/8 and P is the period of the sequence (P = 2**22), is
C still of the form L*P/8.  In particular we find the I-th random
C number multiplied by 2**22 is given by
C
C I    =  0  1*P/8  2*P/8  3*P/8  4*P/8  5*P/8  6*P/8  7*P/8  8*P/8
C
C RAND =  0  5*P/8  2*P/8  7*P/8  4*P/8  1*P/8  6*P/8  3*P/8  0
C
C thus the 4*P/8 = 2097152 random number is 2097152/2**22.
C
C     Several multipliers have been subjected to the spectral test
C (see Knuth, p. 82).  Four suitable multipliers roughly in order of
C goodness according to the spectral test are
C
C    3146757 = 1536*2048 + 1029 = 2**21 + 2**20 + 2**10 + 5
C    2098181 = 1024*2048 + 1029 = 2**21 + 2**10 + 5
C    3146245 = 1536*2048 +  517 = 2**21 + 2**20 + 2**9  + 5
C    2776669 = 1355*2048 + 1629 = 5**9  + 7**7  + 1
C
C     In the table below log10(NU(I)) gives roughly the number of
C random decimal digits in the random numbers considered I at a time.
C C is the primary measure of goodness.  In both cases bigger is
C better.
C
C                  log10 NU(I)                    C(I)
C       A        I=2  I=3  I=4  I=5     I=2  I=3  I=4  I=5
C
C    3146757     3.3  2.0  1.6  1.3     3.1  1.3  4.6  2.6
C    2098181     3.3  2.0  1.6  1.2     3.2  1.3  4.6  1.7
C    3146245     3.3  2.2  1.5  1.1     3.2  4.2  1.1  0.4
C    2776669     3.3  2.1  1.6  1.3     2.5  2.0  1.9  2.6
C   best
C    possible    3.3  2.3  1.7  1.4     3.6  5.9  9.7  14.9
C
C   Input argument --
C         R ---  if R=0.,    the next random number of the
C                            sequence is generated.
C                if R.LT.0., the last generated number
C                            will be returned for possible
C                            use in a restart procedure.
C                if R.GT.0., the sequence of random numbers
C                            will start with the seed R mod 1.
C                            This seed is also returned as the
C                            value of rand provided the arithmetic
C                            is done exactly.
C
C   Output value --
C         RAND - a pseudo-random number between 0. and 1.
C
C   Notes:
C        The value of R is left unchanged by this routine.
C        IA1 and IA0 are the hi and lo parts of A.  IA1MA0 = IA1 - IA0.
C
C***  References: see above.
C***  Routines called:  none
C***  End prologue  RAND
C
C     Declaration of all variables:
      REAL*4 RAND, R
      INTEGER*4 IA1, IA0, IA1MA0, IC, IX1, IX0, IY1, IY0
      LOGICAL FIRST, NOPRNT
C
C     If NOPRNT = .TRUE., the message after the first executable
C     statement will not be printed.
      PARAMETER (NOPRNT = .TRUE.)
C
C     Define constant terms:
      PARAMETER (IA1    = 1536,
     &           IA0    = 1029,
     &           IA1MA0 = IA1 - IA0,
```

```
     &           IC     = 1731       )
C
CC    Save internal value of seed for subsequent calls where
C     R is .le. 0.
         SAVE IX1, IX0
C
CC    Default definition of internal values of seed.
C     (Used if RAND not initially called with R .gt. 0).
      DATA IX1, IX0 /0, 0/
C
      DATA FIRST /.TRUE./
C
C     First executable statement in RAND.
      IF(FIRST .AND. .NOT. NOPRNT) THEN
         FIRST = .FALSE.
         WRITE(*,*) 'Using function RAND (version: 1988-03-09).'
      END IF
C
      IF(R .EQ. 0.0) THEN
C        R is zero so calculate the next random number of the
CC       sequence from:
CCC
CCC         A*X = 2**22*IA1*IX1 + 2**11*(IA1*IX1 + (IA1-IA0)*(IX0-IX1)
CCC                + IA0*IX0) + IA0*IX0
CCC
CCC      where 2**22 = 4194304, and 2**11 = 2048.
C
            IY0  = IA0*IX0
            IY1  = IA1*IX1 + IA1MA0*(IX0-IX1) + IY0
            IY0  = IY0 + IC
            IX0  = MOD(IY0, 2048)
            IY1  = IY1 + (IY0-IX0)/2048
            IX1  = MOD(IY1, 2048)
C
      ELSE IF(R .GT. 0.0) THEN
C        R is positive and non-zero.
CC       Start sequence with external seed R.
            IX1 = AMOD(R,1.)*4194304. + 0.5
            IX0 = MOD (IX1, 2048)
            IX1 = (IX1-IX0)/2048
      END IF
C
CC    Determine a pseudo-random number from the values of the internal
C     seeds IX1 and IX0
      RAND = IX1*2048 + IX0
      RAND = RAND / 4194304.
C
CC
      RETURN
      END
```